Problems in Coding for Multi-Dimensional Storage Devices [1]

Paul H. Siegel

Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407

Abstract — **Advances in the technologies underlying page-oriented data storage systems, such as holographic and 2-photon optical memories, have created a need for 2-dimensional coding and detection methods. This talk will give an overview of some simple 2-dimensional storage channel models, some modulation and channel coding methods that have been proposed for them, and some of the many remaining open problems pertaining to the design and analysis of codes and detectors for multi-dimensional storage systems.**

I. Introduction

There are a number of page-oriented digital data recording technologies that are under active investigation for use in future high performance storage systems [5],[8]. Memory devices based upon these technologies have the potential to offer very high data densities. By employing parallel transfer of input and output pages, they also promise to dramatically increase data access rates. The development of modulation, coding, and detection methods that exploit the inherent multi-dimensionality of these systems presents a number of interesting theoretical and practical challenges. The purpose of this talk is to provide: an introduction to some simple information-theoretic channel models for these systems; an overview of problems in constrained coding, error control, and detection for these channels; and a partial survey of some recent relevant results.

II. Channel Characteristics

The 2-dimensional recording channels considered here may be modeled as a noisy channel that accepts as input a 2-dimensional array of bits and produces at the output a distorted version of the input array in the form of a 2-dimensional array with real-valued entries. The channel may behave like a spatial low-pass filter, distoring high spatial frequencies and inducing intersymbol interference. Defects in the recording medium and other sources of distortion may lead to 2-dimensional burst noise. These characteristics motivate some of the coding and detection methods described below.

III. Modulation Constraints and Codes

There are a number of input-array constraints that have been proposed to improve the performance of the recording and readback process of 2-dimensional optical memories. The extension of the 1-dimensional theory of constrained codes to higher dimensions poses a number of challenges in both the analysis of the constrained systems and the construction of efficient, practical encoding and decoding algorithms.

Among the modulation constraints considered have been conservative array constraints, in which each row and column of a finite array must have at least a prespecified number of transitions from 0 to 1 or 1 to 0 [11],[10]. Another class of constrained arrays that have been studied are DC-free arrays, where each row and column must be balanced [10]. Codes that generate runlength-limited binary arrays, constraining the runlengths of 0's and 1's in rows and columns, have also been investigated [1], [4].

IV. Error Control

In order to compensate for 2-dimensional burst noise and modulation decoder error propagation, error-control coding schemes must cope with clustered errors. There are well-known techniques for correcting certain types of cluster errors. Recently, several new codes and 2-dimensional interleaving techniques have been proposed; see, for example [6], [11], [3], [2].

V. Equalization and Detection

Both holographic and 2-photon optical memories have been modeled as 2-dimensional intersymbol interference (ISI) channels, representable as 2-dimensional digital filters. The extension of detection and coded-modulation methods that have been developed for 1-dimensional, input-constrained ISI channels to their multi-dimensional counterparts remains largely unexplored and offers a number of challenging problems [9], [7].

References

[1] J. Ashley and B. Marcus, "Two-dimensional low-pass filtering codes," submitted to *IEEE Trans. Commun.*, Oct. 1996.

[2] M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 529-542, Mar. 1996.

[3] M. Blaum, J. Bruck, and A. Vardy, "Interleaving schemes for multi-dimensional cluster errors," *IEEE Trans. Inform. Theory*, to appear.

[4] T. Etzion, "Cascading methods for runlength-limited arrays," *IEEE Trans. Inform. Theory*, vol. 43, no. 1, pp. 319-324, Jan. 1997.

[5] J. Heanue, M. Bashaw, and L. Hesselink, "Volume holographic storage and retrieval of digital data," *Science*, vol. 265, pp. 749-752, 1994.

[6] J. Heanue, M. Bashaw, and L. Hesselink, "Channel codes for digital holographic data storage," *J. Opt. Soc. Am. A*, vol. 12, pp. 2432-2439, 1995.

[7] J. Heanue, K. Gurkan, and L. Hesselink, "Signal detection for page-access optical memories with intersymbol interference," *Appl. Opt.*, vol. 35, no. 14, pp. 2431-2438, May 1996.

[8] S. Hunter, F. Kiamilev, S. Esener, D. Parthenopoulos, and P.M. Rentzepis,"Potentials of two-photon based 3D optical memories for high performance computing," *Appl. Opt.*, vol. 29, pp. 2058-2066, 1990.

[9] B. Olson and S. Esener,"Partial response precoding for parallel-readout optical memories," *Opt. Lett.*, vol. 19, pp. 661-663, 1993.

[10] R. Talyansky, T. Etzion, and R. Roth, "Efficient code constructions for certain two-dimensional constraints," extended abstract, 1997.

[11] A. Vardy, M. Blaum, P. Siegel, and G. Sincerbox, "Conservative arrays: multi-dimensional modulation codes for holographic recording," *IEEE Trans. Inform. Theory*, vol. 42, pp. 227-230, Jan. 1996.

[1] This work was supported by The National Science Foundation under Grant NCR-9612802.

Codes correcting phased burst erasures

Osnat Keren and Simon Litsyn
Tel-Aviv University
Department of Electrical Engineering – Systems
Ramat-Aviv 69978
Tel-Aviv, Israel

Abstract — **We introduce a family of binary array codes of size $t \times n$, correcting multiple phased burst erasures of size t. The codes achieve maximal correcting capability, i.e. being considered as codes over $GF(2^t)$ they are MDS. The length of the codes is $n = \sum_{\ell=1}^{L} \binom{t}{\ell}$, where L is a constant or is slowly growing in t. The complexity of encoding and decoding is proportional to $rnmL$, where r is the number of correctable erasures, and m is the smallest number such that $2^t = 1$ modulo m. This compares favorably with the complexity of decoding codes obtained from the shortened Reed-Solomon codes having the same parameters.**

I. Introduction

Let $t \times k$ bits of information be encoded in a $t \times n$ bit array, $n > k$. Due to some reasons the data stored in several columns can be lost or corrupted. We assume that we know in which columns it has happened. Our purpose is to reconstruct the missing data.

A binary $C(t, n, r)$ code for $n \leq 2^t - 1$ can be obtained by shortening a Reed-Solomon (RS) code over $GF(2^t)$. A codeword can be regarded as a vector of n symbols belonging to $GF(2^t)$, where each symbol is a t-bit column. With slight abuse of terminology, we say that n is the length of the array code.

The conventional decoding procedure consists of calculating syndrome followed by computing the values of erasures. This can be done using the Forney algorithm. All the operations in the algorithm are implemented in $GF(2^t)$ and its complexity is proportional to rn field operations or $rnt \log t \log \log t$ bit operations [4]. Since we need calculation of the syndrome, decoding cannot be simpler than being proportional to rnt bit operations. So, the question is if it is possible to design codes achieving the maximum possible erasure-correcting capability, and having complexity of decoding proportional to rnt. A class of such codes was constructed by Blaum and Roth [1] (for earlier results see [1, 2] and references therein). Having complexity of decoding proportional to rnt, the Blaum-Roth codes are of length at most $t + 1$, and t is of the form $p - 1$ for some prime p. However, these constraints on the code parameters are quite restrictive.

Here we propose a new class of $C(n, t, r)$ array codes with maximal erasure-correcting capability. The length of the codes is $n \leq \sum_{\ell=1}^{L} \binom{t}{\ell}$, for some $L \leq t$. If L is chosen to be constant independent of t, the codes have length proportional to t^L. Let m be the minimal number such that $m > t, m | 2^t - 1$. Then, complexity of decoding the proposed codes is of order $rnmL$, i.e. if m is close to t and L is constant, it is proportional to rnt. If $L < \log t \log \log t$, the decoding complexity is still better than for general shortened RS codes. If $L = 1$ and m is prime then t equals $m - 1$ and our construction coincides with the one of Blaum-Roth for $n \leq t$.

The basic idea is that we use a shortened RS code over the field $GF(2^t)$ defined by a **nonprimitive** polynomial $M(x)$, having a root α of order $m > t$. It is desirable to have m as close as possible to t. Considering the elements of the field as polynomials in α, we pick only those columns of the parity check matrix of the RS code that correspond to the elements with at most L nonzero coefficients. The essential simplification is achieved by implementing the computations in the ring defined modulo the polynomial $x^m - 1$ where multiplication by a power of α turns out to be a cyclic shift.

II. Construction

Let the field $F = GF(2^t)$ be defined modulo a **nonprimitive** irreducible polynomial $M(x)$ of degree t. Denote by α a root of $M(x)$, α has order $m > t$. Hence, α is also a root of $x^m - 1 = M(x)g(x)$. We define the enveloping polynomial ring R modulo $x^m - 1$. To avoid confusion between the elements of F and the elements of R, we use Greek letters for F and bold Latin letters for R.

Let $B = \{\alpha^i, 0 \leq i < t\}$, the elements of B are linearly independent over $GF(2)$ and can be used as a basis for F. Every $\theta \in F$ can be represented as $\sum_{k=0}^{t-1} \theta^{(k)} \alpha^k$, $\theta^{(i)} \in GF(2)$.

Let $\mathbf{a} = \sum_{i=0}^{m-1} \mathbf{a}^{(i)} \alpha^i \in R$. We define $\mathcal{T} : F \to R$, a mapping from the field to the ring, namely $\mathbf{a} = \mathcal{T}(\theta)$, where

$$\mathbf{a}^{(i)} = \begin{cases} \theta^{(i)} & 0 \le i < t \\ 0 & t \le i < m. \end{cases} \tag{1}$$

The inverse $\mathcal{T}^{-1}$ of $\mathbf{a}$ is obtained by computing $\mathbf{a}$ modulo M.

Let us define the rotation operator $\mathcal{R}_\ell(\mathbf{a}) = \mathbf{b}$, where $\mathbf{b}^{(i)} = \mathbf{a}^{(i-\ell)}$ and the indices are taken modulo m. Multiplying an element $\mathbf{a} \in R$ by α is simply a cyclic shift of the coefficients of $\mathbf{a}$, and therefore $\alpha^l \mathbf{a} = \mathcal{R}_\ell(\mathbf{a})$. Hence, multiplication of two elements of R can be implemented in Lm binary XOR operations, where L is the number of nonzero coefficients of one of the elements.

Now we are ready to describe the codes. For some positive integer L let the set $B_L \subset F$ be the collection of all possible $N = \sum_{l=1}^{L} \binom{t}{l}$ linear combinations of at most L elements out of the t elements of the set B. In other words B_L consists of all the polynomials having at most L nonzero coefficients, $|B_L| = N$.

A codeword $\underline{c} \in C(t,n,r)$ is a vector over F of length $n \le N$ (evidently, $\underline{c}$ can be also regarded as a $t \times n$ array over $GF(2)$). The code is defined by the following parity check matrix H :

$$H = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ \theta_0 & \theta_1 & \theta_2 & \dots & \theta_{n-1} \\ \theta_0^2 & \theta_1^2 & \theta_2^2 & \dots & \theta_{n-1}^2 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \theta_0^{r-1} & \theta_1^{r-1} & \theta_2^{r-1} & \dots & \theta_{n-1}^{r-1} \end{pmatrix}$$

$$= (\underline{h}_0, \underline{h}_1, \dots, \underline{h}_{n-1}). \tag{2}$$

Here $\theta_i \in B_L$, $\theta_i \ne \theta_j$ and $\underline{h}_i \in F^r$ for $i = 0, 1, \dots, n-1$.

Lemma 1 *The array code $C(t,n,r)$ is MDS code over F with minimum distance $d = r+1$.*

III. Decoding r erasures

Let $\underline{c} = (\beta_0, \beta_1, \dots, \beta_{n-1}) \in C, \beta_i \in F$, be a codeword. Assume that at most r erasures occured, and $\underline{y} = (\vartheta_0, \vartheta_1, \dots, \vartheta_{n-1})$ is the word that equals to $\underline{c}$ except in the coordinates where erasures occurred, the value of $\underline{y}$ in these positions is set to $0 \in F$. Let $\underline{s} = (\xi_0, \xi_1, \dots, \xi_{r-1})^T, \xi_i \in F$, be the syndrome, $\underline{s} = H\underline{y}^T$.

Assume that $\rho \le r$ columns have been erased at the coordinates $0 \le i_0 < i_1 < \dots < i_{\rho-1} \le (n-1)$. Let $\beta_{i_l} \in F$ be the value of the erased column in the position i_l. The elements $\{\beta_{i_l}\}$ satisfy the set of r linear equations over F,

$$\sum_{l=0}^{\rho-1} \beta_{i_l} \underline{h}_{i_l} = \underline{s}. \tag{3}$$

By Forney, each β_{i_l} satisfies:

$$\left(\prod_{k=0, k\ne l}^{\rho-1} (\theta_{i_k} + \theta_{i_l}) \right) \beta_{i_l} = \Gamma_{l,\rho-1}, \tag{4}$$

where $\Gamma_{l,\rho-1} \in F$ is the coefficient of $x^{\rho-1}$ in the polynomial $\Gamma_l(x) = \Omega(x)/(1-\theta_{i_l}x)$. The polynomial $\Omega(x)$ is the erasure-evaluator polynomial, $\Omega(x) = S(x)\Lambda(x)$ (mod x^ρ), where $S(x)$ is the syndrome polynomial, and $\Lambda(x)$ is the erasure-locator polynomial defined as follows

$$S(x) = \textstyle\sum_{k=0}^{\rho-1} \xi_i x^i \quad , \quad \Lambda(x) = \prod_{k=0}^{\rho-1}(1 - \theta_{i_k} x).$$

To simplify the computation, we perform it in the ring R, where the complexity of multiplying an element is determined by the number of terms it has. The algorithm for decoding $\rho \le r$ erasures has five steps (detailed description of the decoding algorithm is given in [3]):

Step 1: Syndrome calculation in the ring R.
Step 2: Calculation of $\Omega(x)$ once for all erasures.
Step 3: Computation of $\Gamma_l(x)$ for each erasure.
Step 4: Calculation of the product γ_l, defined by the left hand side of (4).
Step 5: Extract β_{i_l} by using the Euclidean Algorithm.

Let $t\mathcal{L}(t)$ be the complexity of multiplication in F. The best known estimate for $\mathcal{L}(t)$ is $O(\log t \log\log t)$, see e.g. [4]. The following table, compares the complexities of the proposed algorithm and the Forney algorithm.

description	Forney	Algorithm
syndrome calculation	$rnt\mathcal{L}(t)$	$rnLm$
$\Omega(x)$ calculation	$r^2t\mathcal{L}(t)$	r^2Lm
$\Gamma_{l,\rho-1}$ calculation	$r^2t\mathcal{L}(t)$	r^2Lm
γ_l calculation	$r^2t\mathcal{L}(t)$	r^2Lm
Extract β_{i_l}	$rt\log t\mathcal{L}(t)$	$rt\log t\mathcal{L}(t)$

References

[1] M. Blaum and R. M. Roth, New Array Codes for Multiple Phased Burst Correction, *IEEE Trans. Inform. Theory*, vol.39, No.1, pp. 66–77, 1993.

[2] S. J. Hong and A. Patel, A general class of maximal codes for computer applications. *IEEE Trans. Comput.*, vol.21, No.12, pp.1322-1331, 1972.

[3] O.Keren and S.Litsyn, Codes correcting phased burst erasures, submitted to *IEEE Trans. on Information Theory*.

[4] I. E. Shparlinski, *Computational and Algorithmic Problems in Finite Fields*, Kluwer, 1992.

Constructing DC-Free Runlength-Limited Block Codes

Khaled A. S. Abdel-Ghaffar[1]
Department of Electrical
and Computer Engineering
University of California
Davis, CA 95616
USA
ghaffar@ece.ucdavis.edu

Jos H. Weber
Delft University of Technology
Department of Electrical Engineering
Telecommunications and Traffic-Control
Systems Group
P.O.Box 5031
2600 GA Delft, The Netherlands
weber@et.tudelft.nl

Abstract — **A general scheme for DC-free runlength-limited block codes is considered. Constructions leading to maximal code rates are specified for practical runlength-limited constraints.**

I. Introduction

Many modulation systems used in magnetic and optical recording are based on binary runlength-limited codes that are DC-free, i.e., whose power spectral densities exhibit a null at zero-frequency. A string of bits is said to be runlength-limited if the number of '0's between two consecutive '1's is bounded between a certain minimal value d and a certain maximal value k. Such sequences are also called (d,k) constrained. The lower runlength constraint is imposed to reduce intersymbol interference, while the upper runlength constraint is imposed to maintain synchronization. The null at zero-frequency is imposed in order to prevent the recorded data from being affected by the low-frequency signals used to control the servo systems. The null is also required if the data is processed by high-frequency filters that remove low-frequency noise. Furthermore, to prevent signal distortion when AC-coupling is used, it is often necessary to avoid charge accumulation by requiring the data to be DC-free. For an excellent introduction to the theory of DC-free runlength-limited sequences and its applications we refer the reader to [2]. The following notation will be used. Let d, k, l, r, and n be integers satisfying $0 \leq d < k$ and $0 \leq l, r \leq k$ unless explicitly stated otherwise. A *dklr*-sequence is a binary sequence starting with at most l '0's, ending with at most r '0's, and having at least d and at most k '0's between any two consecutive '1's. In case there is no upper runlength constraint, we say $k = \infty$. Let $\hat{\mathcal{M}}_{dklr}(n)$ be the set of nonzero *dklr*-sequences of length n and $\hat{M}_{dklr}(n)$ be their number. We denote by 0^i, where i is a nonnegative integer, the sequence of i '0's.

We consider a block coding scheme which converts data into a DC-free, (d,k) constrained, bit stream. In this scheme messages, from a set $\mathcal{A}$ of A messages, are one-to-one mapped to *dklr*-sequences of length n. Between any two of such sequences we insert b appropriate merging bits. These merging bits are ignored by the decoder. In order to keep the encoding and decoding simple, we choose both n and b to be fixed. Notice that $N = n+b$ bits are assigned to each message and the code rate is therefore $R = (\log_2 A)/N$. The merging bits should ensure that the following two conditions are satisfied:

- The (d,k) runlength constraint,
- The DC-free constraint.

To satisfy the first condition, the merging bits are chosen based on the numbers of recorded/transmitted consecutive '0's before and after the merging bits. To simplify the scheme, we will assume that no message is mapped into the all-zero *dklr*-sequence if such sequence exists. With this assumption, the numbers of recorded/transmitted consecutive '0's before and after the merging bits equal the number of trailing '0's in the previous *dklr*-sequence and the number of leading '0's in the next *dklr*-sequence, respectively.

To satisfy the second condition, the merging bits should be chosen in a way that keeps the running digital sum (RDS) bounded [2], [4]. This is accomplished by providing two choices with differing weight parities for the sequence of merging bits between any two *dklr*-sequences, i.e., an even-weight sequence and an odd-weight sequence. These two sequences depend only on the numbers of trailing and leading '0's in the previous and in the next *dklr*-sequences, respectively. To keep the RDS bounded, a sequence of merging bits among these two alternatives is chosen based on a simplified version of an algorithm presented in [1]. This choice depends on the RDS and the total number of '1's of the recorded/transmitted sequence just before the merging bits and on the next two *dklr*-sequences.

We consider only (d,k) constraints with capacities at least equal to 0.5 since other constraints are of little practical interest. It can be verified that the capacity of a (d,k) constraint is at least equal to 0.5 if and only if $d = 0 \wedge k \geq 1$, $d = 1 \wedge k \geq 3$, or $d = 2 \wedge k \geq 7$.

For given d, k, and A, our goal is to maximize the code rate when encoding sequences of messages from a set of size A into DC-free (d,k) constained sequences using the aforementioned scheme. To maximize the code rate for given l and r, the length of the *dklr*-sequences, n, should be the smallest positive integer n for which $\hat{M}_{dklr}(n) \geq A$. Furthermore, the number of merging bits, b, should be the smallest integer such that for any two nonzero *dklr*-sequences, there are even-weight and odd-weight sequences of length b and each of these sequences can be used as merging bits between the two *dklr*-sequences to maintain the (d,k) constraint. Let n_{lr} and b_{lr} be these smallest n and b, respectively. Thus, the maximal code rate for given l and r equals $(\log_2 A)/(n_{lr} + b_{lr})$. To maximize the code rate over all values of l and r, we have to search for a choice of l and r for which $n_{lr} + b_{lr}$ is minimal. To this end, note the following with regard to this choice. On one hand, large values of l and r may lead to small values of n_{lr} since the number of *dklr*-sequences of length n is nondecreasing in l and r. On the other hand, small values of l and r may require less merging bits b_{lr}. This dilemma plays a central part in maximizing the code rate of the scheme. An option for l and r that maximizes the code rate is

[1] This author was supported by NSF Grant NCR 96-12354.

called optimal.

II. Summary of Results

Depending on the values of d, k, and A, we present optimal options for l and r. We also specify the even-weight sequence β_e and the odd-weight sequence β_o of minimal length b_{lr} that can be used for cascading two given $dklr$-sequences. In general, β_e and β_o depend on the number s of trailing '0's in the preceding $dklr$-sequence and the number t of leading '0's in the following $dklr$-sequence.

Case I $k = \infty$:
The option $l = r = \infty$ is optimal. For this option $b_{lr} = 2d + 1$, $\beta_e = 0^{2d+1}$, and $\beta_o = 0^d 1 0^d$.

Case II $0 = d < k < \infty$:
The option $l = \lfloor (k-1)/2 \rfloor$ and $r = \lceil (k-1)/2 \rceil$ is optimal. For this option $b_{lr} = 1$, $\beta_e = 0$, and $\beta_o = 1$.

Case III $d = 1 \wedge k = 3$:
The option $l = r = 2$ is optimal. For this option $b_{lr} = 5$, $\beta_e = 01010$, and

$$\beta_o = \begin{cases} 00010 & \text{if } s = 0, \\ 01000 & \text{if } s \geq 1 \text{ and } t = 0, \\ 10101 & \text{if } s \geq 1 \text{ and } t \geq 1. \end{cases}$$

Case IV $4 \leq 3d + 1 = k \wedge A \geq d + 2$:
The option $l = k - d - 1$ and $r = k - d$ is optimal. For this option $b_{lr} = 2d + 2$. The sequences β_e and β_o are given in Table 1.

Case V $1 \leq d \leq 2 \wedge 3d+2 \leq k < \infty$ or $4 \leq 3d+1 = k \wedge A \leq d+1$:
The option $l = r = k - 2d - 1$ is optimal. For this option $b_{lr} = 2d + 1$. The sequences β_e and β_o are given in Table 2.

III. Conclusion and Discussion

We considered a general scheme to construct block codes that are both DC-free and (d, k) constrained. In this scheme, messages are mapped to distinct $dklr$-sequences and merging bits are added between consecutive $dklr$-sequences to ensure that both constraints are satisfied. We determined options for l and r that maximize the code rate for all (d, k) constraints with capacity 0.5 or greater. In general, the code rates tend to the capacity of the (d, k) constraint as the number of messages A increases.

Recently, Lin and Liu [3] proposed DC-free $(1, k)$ constrained block codes with $k \geq 4$ that belong to the general scheme considered here. For these codes, three merging bits are inserted between $1klr$-sequences, where $l = r = k - 3$. Our results indicate that this option of l and r is optimal except in the case $k = 4$ and $A \geq 3$.

In an earlier paper [5], the authors presented a general theory for constructing block codes that are (d, k) constrained but not necessarily DC-free. The construction is based on mapping messages to $dklr$-sequences and inserting merging bits between any two consecutive sequences. Only one sequence of merging bits is provided between any two $dklr$-sequences to maintain the (d, k) constraint. It is shown in [5] that for $d \geq 1$, $k < \infty$, and k sufficiently large compared to d, which is the case in most applications, maximal rate codes are obtained by taking $l = r = k - d$ and inserting merging sequences of d bits. On the other hand, the results presented here show that if we want the code to be DC-free, then maximal rate codes are obtained by taking $l = r = k - 2d - 1$ and inserting merging sequences of $2d + 1$ bits. By requiring the DC-free condition, not only we need more merging bits but we may also need longer $dklr$-sequences since the number of such sequences with $l = r = k - 2d - 1$ is typically less than the number of sequences with $l = r = k - d$.

References

[1] D. Coppersmith and B.P.Kitchens, "Run-length limited code without dc level," U.S. Patent 4675650, Jun. 23, 1987.

[2] K.A.S. Immink, *Coding Techniques for Digital Recorders*, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.

[3] Y. Lin and P.-H. Liu, "A construction technique for charge constrained $(1, k \geq 4)$ codes," *IEEE Trans. Magn.*, vol. 31, pp. 3081–3083, Nov. 1995.

[4] G. L. Pierobon, "Codes for zero spectral density at zero frequency," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 435–439, Mar. 1984.

[5] J.H. Weber and K.A.S. Abdel-Ghaffar, "Cascading runlength-limited sequences," *IEEE Trans. Inform. Theory*, vol. IT-39, no. 6, pp. 1976–1984, Nov. 1993.

Tab. 1: Merging sequences for case IV

s, t	β_e	β_o
$s + t \leq k - 2d - 2$	0^{2d+2}	$0^{d+1}10^d$
$s + t \geq k - 2d - 1 \wedge s \leq d - 1 \wedge t \leq d - 1$	$0^t 10^d 10^{d-t}$	$0^t 10^{2d-t+1}$
$s + t \geq k - 2d - 1 \wedge s \leq d - 1 \wedge t \geq d$	$0^d 10^d 1$	$0^{2d+1}1$
$s + t \geq k - 2d - 1 \wedge s \geq d \wedge t \leq k - 2d - 1$	$10^d 10^d$	10^{2d+1}
$s + t \geq k - 2d - 1 \wedge s \geq d \wedge t \geq k - 2d$	$10^{2d}1$	$0^{t+2d-k+1}10^{k-t}$

Tab. 2: Merging sequences for case V

s, t	β_e	β_o
$s + t \leq k - 2d - 1$	0^{2d+1}	$0^d 10^d$
$s + t \geq k - 2d$ and $s \leq d - 1$	$0^{d-s}10^d 10^{s-1}$	$0^{d-s}10^{d+s}$
$s + t \geq k - 2d$ and $s \geq d$	$10^d 10^{d-1}$	10^{2d}

An Upper Bound for Correct Path Loss Probability in the M-Algorithm over ISI Channels

Lei WEI[1]

Department of Engineering, FEIT, The Australian National University (ANU), ACT0200, Australia

Abstract — **We present an upper bound on the correct path loss probability for the bread-first limited over finite ISI channels. Several properties are proved and an algorithm to compute approximately the upper bound is given.**

I. Introduction

THE famous Viterbi Algorithm (VA) has been widely applied in digital communications. The complexity of the VA (in term of the number of states) grows with the number of states. Therefore, in practice, when the number of states becomes large, it is desired to reduce the receiver complexity but still maintain a near optimum error performance, especially for asymptotic cases. Many algorithms have been proposed to reduce the complexity of the VA [2]-[11] . The breadth-first algorithms are especially promising [2]-[8]. For interference channels, the breadth-first algorithms can often achieve a near optimum performance with a very low complexity [6]-[7].

In [10], we have shown that the bit error probability in the breadth first limited search detection (i.e., the well-known M-algorithm) over a finite interference channel can be bounded from above by the sum of three terms: the upper bound on the error probability in the Viterbi algorithm detection given in [1] and two upper bounds on the two types of the error probability caused by the correct path loss event. Error propagation and correct path loss (CPL) are two important events affecting the performance of the M-algorithm. In this paper we will focus on the computation of the upper bound (given in equation 1) of correct path loss probability. The concept of the Vector Euclidean Distance (VED) and several properties were given in [11] to simplify the procedure of computing the minimum VED. In this paper, we will also introduce new properties and rigorously prove them. Then, based on these properties, we will present an algorithm which can be used to compute the upper bound.

In [10], we have shown that the probability of CPL is bounded by

$$P_{CPL} \leq \sum_{z \in Z} 2^{-w(z)} Q(\sqrt{\frac{d_\ell^2(z) E_b}{N_0}}) \qquad (1)$$

where $Q(x) = \frac{1}{2\pi} \int_x^\infty exp(-\xi^2/2) d\xi$, E_b and N_0 denote the bit energy and power spectral density of Gaussian noise respectively, Z denotes a set comprised of all possible sets of M+1 paths (one correct path and M error paths), $w(z)$ denotes the maximum Hamming distance between the correct path and any error path, $d_\ell^2(z)$denotes the VED of set z, which has been defined in [11], [10] and [12].

II. PROPERTIES

The following properties are important for establishing an efficient algorithm to compute the upper bound.

Property 1. If set z is comprised of paths $\{a_0, \ldots, a_M\}$ and z_+ is comprised of paths $\{a_0, \ldots, a_M, a_{M+1}\}$, then $d^2_{\ell, z_+}(M+1) \geq d^2_{\ell,z}(M)$, i.e., the VED of z_+ is no less than that of z.

Property 2. Both VED and P_{CPL} are non-decreasing functions of M.

Property 3. The VED of set z is no less than the Euclidean distance between any two paths of M paths.

Let a_1 and a_2 be two sequences of M-ary data symbols that form an error event starting at 0 (without loss of generality) and ending at kT. Let δ be the minimum nonzero increment of the (normalized and squared) Euclidean distance defined in equation 2,

$$\delta = \underset{(a_1,a_2) \in allerrorevents}{min} \quad \underset{0 \leq N \leq k-1}{inf} \{r > 0; \qquad (2)$$

$$r = \tfrac{1}{2E_b} \int_{NT}^{(N+1)T} [s(t,a_1) - s(t,a_2)]^2 dt\}$$

where $s(t, a_1)$ denotes the transmitted signal related to sequence a_1 and T is the symbol duration.

Let the maximum interval in which an error event has no distance increment be $k_{max}T$, where k_{max} is defined in equation 3.

$$k_{max} = \underset{(a_1,a_2) \in allerrorevents}{max} \quad \underset{0 \leq N \leq k}{sup} \qquad (3)$$

$$\{k_1 = 1, \ldots, k-N; \int_{NT}^{(N+k_1)T} [s(t,a_1) - s(t,a_2)]^2 dt = 0\}$$

Property 4: If $k_{max} < \infty$ or $\delta \neq 0$, then there is a shortest split-interval, N_s, such that a minimization over a longer interval will not produce a smaller value of minimum VED.

III. AN ALGORITHM

An algorithm will be given to compute the upper bound in equation 1 up to a threshold H, i.e., sum of all sets whose VED is less than H. The larger the threshold, the closer the upper bound can be approximated, but more computational effort is required.

Step 1: Firstly, we specify the threshold value of VED (H) which we are interested. After selecting H, we need to find all possible error vectors whose Euclidean distances are less than H. We do not consider the error vectors with a distance larger than H according to property 3.

Step 2: Start from the M=2 case. Form a set of 3 paths from the survived error vectors, then exam its VED. If its VED is larger than H, then the set is discarded. Otherwise, the set is survived as one of survived three path sets.

Step 3: Increase M by 1. Form a set of M+1 paths from the survived M path sets, then exam its VED. If its VED is larger than H, then the set is discarded. Otherwise, the set is survived as one of survived M+1 path sets.

Step 4: Repeat step 3 until no survived sets.

The key concept is to form a set of M+1 paths from the survived

[1] This work was supported in part by ATERB under Grant no. N063/405

M path sets. It was found that many survived error vectors could not form any 3 path set, thus these sets can be discarded before computing VED.

Example: Let us consider an ISI channel with $f(D) = 0.648+0.5657D+0.4243D^2+0.2828D^3$. Its minimum Euclidean distance is 1.09. The values of VED for M=1, 2, 3 are 0.839, 1.429 and 1.888, respectively. For H=2.0, there are 55 error vectors, 21 three path sets and 6 four path sets survived. The bounds on the CPL probabilities and the simulation results are presented in figure 1.

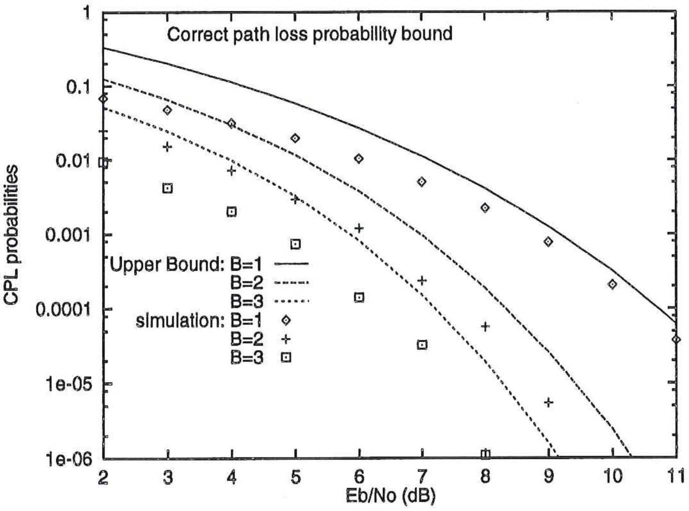

Fig. 1: CPL probabilities for the ISI channel with B=1, 2, 3.

IV. CONCLUSION

We have presented several properties of the VED and given an algorithm to compute an upper bound on correct path loss probability. An example is also given. It is worth mentioning that all the results in this paper and [10] may be extended to error control coding, which may lead us to design better error control codes for the M-algorithm.

REFERENCES

[1] D.G. Forney, Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Trans. Inform. Theory*, Vol.IT-18, pp.363-378, No.3, May., 1972.

[2] T. Aulin, "A Fractional Viterbi-Type Trellis Decoding Algorithm," *Proceeding of IEEE ISIT'86*, Oct. 1986, Michigan, USA, p.41.

[3] G. J. Pottie and D. P. Taylor, "A Comparison of Reduced Complexity Decoding Algorithms for Trellis Codes," *IEEE J. Select. Areas Commun.*, Vol.7, pp. 1369-1380, Dec. 1989.

[4] J. B. Anderson and S. Mohan, "Sequential Coding Algorithms: A Survey and Cost Analysis," *IEEE Trans. Commun.*, Vol. 32, pp. 169-176, Feb. 1984.

[5] J. B. Anderson, "Limited Search Trellis Decoding of Convolutional Codes," *IEEE Trans. Inform. Theory*, Vol. 35, pp. 944-955, Sept. 1989.

[6] A. Duel-Hallen and C. Heegard, "Delayed Decision-Feedback Sequence Estimation," *IEEE Trans. Commun.*,Vol. 37 pp. 428-436, May 1989.

[7] L.Wei, L.K.Rasmussen and R.Wyrwas, "Near Optimum Tree-Search Detection Schemes for Bit Synchronous Multiuser CDMA Systems Over Gaussian and Two-path Rayleigh Fading Channels," *IEEE Trans. on Communication*, to appear.

[8] S.T. Simmons, "Breadth-First Trellis Decoding with Adaptive Effort," *IEEE Trans. Commun.*, Vol. 38, pp. 3-12, Jan., 1990.

[9] J. B. Anderson and E. Offer, "Reduced-State Sequence Detection with Convolutional Codes," *IEEE Trans. Inform. Theory*, Vol.IT-40, pp.965-972, No.3, May, 1994.

[10] L. Wei and H. Qi, "An Upper Bound on the Error Probability in Breadth First Limited Search Detection over Finite Interference Channels," submitted to *IEEE Trans. Inform. Theory*, 1996.

[11] T. Aulin, "Breadth First Maximum Likelihood Sequence Detection," submitted to *IEEE Trans. Inform. Theory*, 1992.

[12] C. Schlegel, Trellis coded Modulation, *IEEE Press*, 1995.

Efficient Encoding Algorithm for Third-Order Spectral-Null Codes[1]

Vitaly Skachek Tuvi Etzion Ron M. Roth[2]
Computer Science Department
Technion, Haifa 32000, Israel.
e-mail: {vitalys, etzion, ronny}@cs.technion.ac.il

Abstract — **An efficient algorithm is presented for encoding unconstrained information sequences into a third-order spectral-null code of length n and redundancy $9\log_2 n + O(\log\log n)$. The encoding can be implemented using $O(n)$ integer additions and $O(n\log n)$ counter increments.**

I. Introduction

Let F be the bipolar alphabet $\{+1, -1\}$. A word $\underline{x} = (x_1, x_2, \ldots, x_n)$ in F^n is a *k-th order spectral-null word* (at zero frequency) if the respective real polynomial $x_1 z + x_2 z^2 + \ldots + x_n z^n$ is divisible by $(z-1)^k$. We denote by $\mathcal{S}(n,k)$ the set of all k-th order spectral-null words in F^n. Any subset $\mathcal{C}$ of $\mathcal{S}(n,k)$ is called a *k-th order spectral-null code* of length n. The concatenation of any l words in $\mathcal{C}$ yields a word in the set $\mathcal{S}(nl,k)$; so, spectral-null codes can be used as block codes with a *redundancy* of $n - \log_2 |\mathcal{C}|$ bits (per block of length n).

The set $\mathcal{S}(n,k)$ is equivalently characterized by

$$\mathcal{S}(n,k) = \left\{ \underline{x} \in F^n : \sum_{j=1}^{n} (j+c)^\ell x_j = 0,\ 0 \le \ell \le k-1 \right\} \quad (1)$$

where c is any real constant (see [4, Ch. 9], [6]).

First-order spectral-null codes are also known by the names *balanced codes*, *zero-disparity codes*, or *DC-free codes*. There is a known efficient encoding algorithm for these codes due to Knuth [3] (see also Al-Bassam and Bose [1]), resulting in codes with redundancy $\log_2 n + O(\log\log n)$, where n is the code length. By 'efficient' we refer to the time (and space) complexity of the encoding, which amounts in Knuth's algorithm to $O(n)$ increments/decrements of a $\lceil \log_2 n \rceil$-bit counter (memory trade-offs allow to reduce the redundancy to $\log_2 n + O(1)$). The redundancy of $\mathcal{S}(n,1)$ is $\frac{1}{2}\log_2 n + O(1)$, and such redundancy can be attained by enumerative coding [4]; in terms of complexity, however, enumerative coding is less efficient than Knuth's algorithm.

For the case $k = 2$, efficient coding algorithms were presented in [6] and [7] that have redundancy of $3\log_2 n + O(\log\log n)$ bits and time complexity that amounts to $O(n)$ additions of $O(\log n)$-bit integers. Enumerative coding already turns out to be impractical for this case [6]. The redundancy of $\mathcal{S}(n,2)$ is known to be $2\log_2 n + O(1)$ [7].

For higher orders k of spectral null, Karabed and Siegel presented in [2] a coding method based upon finite-state diagrams (see also Monti and Pierobon [5]). However, since the rate of their construction is strictly less than 1, the resulting redundancy is linear in the code length n. It follows that for any fixed k and sufficiently large n, this redundancy is significantly larger than the upper bound $O(2^k \cdot \log n)$ on the redundancy of $\mathcal{S}(n,k)$ which is proved in [6] by nonconstructive arguments. A recursive construction is presented in [6] whose redundancy is $O(n^{1-\epsilon_k})$, where $0 < \epsilon_k < 1$ and $\lim_{k\to\infty} \epsilon_k = 0$. Yet, this redundancy is still considerably larger than the actual redundancy of $\mathcal{S}(n,k)$.

In this work, we present an efficient algorithm for encoding unconstrained sequences into a third-order spectral-null code whose redundancy is logarithmic in the code length. More specifically, for code length n, the redundancy is $9\log_2 n + O(\log\log n)$ bits and the encoding complexity is $O(n)$ additions of $O(\log n)$-bit integers and $O(n\log n)$ increments/decrements of $\lceil \log_2 n \rceil$-bit counters.

II. A Third-Order Spectral-Null Encoder

It was shown in [6] that the length of a third-order spectral-null word is divisible by 4; so, the words generated by our encoder will be of length $n = 2h$ for some even integer h. For such an n, we let m be the integer $\lceil \log_2 n \rceil = 1 + \lceil \log_2 h \rceil$. Our encoding scheme will map input words $\underline{y}$ of length $\ge 2h - 6m + 2$ over F, into words $\underline{x} \in \mathcal{S}(2h,3)$ and $\underline{x}' \in \mathcal{S}(3m + O(\log m), 3)$; the concatenation of $\underline{x}$ and $\underline{x}'$, in turn, will form the output third-order spectral-null word.

We will use the definition of $\mathcal{S}(2h,3)$ obtained from (1) by substituting $k = 3$ and $c = -h-1$. It will also be convenient to index the entries of $\underline{x}$ hereafter by $(x_{-h}, x_{-h+1}, \ldots, x_{h-1})$. For a real word $\underline{x}$ we define the *moments* of $\underline{x}$ by

$$\sigma_\ell(\underline{x}) \overset{\text{def}}{=} \sum_{j=-h}^{h-1} j^\ell \cdot x_j\ , \quad \ell = 0, 1, 2, \cdots .$$

Clearly, a word $\underline{x} \in F^n$ is in $\mathcal{S}(2h,3)$ if and only if $\sigma_0(\underline{x}) = \sigma_1(\underline{x}) = \sigma_2(\underline{x}) = 0$.

Our encoding algorithm starts with a word $\underline{x}$ over $F \cup \{0\}$ that contains the input word $\underline{y}$ as a subword, and the remaining entries of $\underline{x}$ are initially set to zero. Next, the algorithm reduces to zero the absolute values of $\sigma_0(\underline{x})$, $\sigma_2(\underline{x})$, and $\sigma_1(\underline{x})$ (in that order), by a sequence of bit negations, bit shifts, and bit swaps, and by assigning values of F to the zero entries. The encoding ends by coding recursively certain counters that were computed in the course of the algorithm, resulting in a word, $\underline{x}'$, which is concatenated with $\underline{x}$ to produce the final output third-order spectral-null word.

The algorithm makes use of the following index sets, all being subsets of $S = \{-h, -h+1, \ldots, h-1\}$:

- $S_{\mathrm{C2}} = \{d_i\}_{i=0}^{2m-8} \cup \{e_i\}_{i=0}^{2m-8}$, where, for $0 \le i \le 2m-10$,

$$(d_i, e_i) = \begin{cases} (-10 \cdot 2^{i/2}, -6 \cdot 2^{i/2}) & \text{for even } i \\ (-9 \cdot 2^{(i+1)/2}, -7 \cdot 2^{(i+1)/2}) & \text{for odd } i \end{cases}$$

[1] This work was supported in part by grant No. 95-522 from the U.S.–Israel Binational Science Foundation, Jerusalem, Israel.

[2] Currently on sabbatical leave, visiting Hewlett-Packard Laboratories, 1501 Page Mill Road, Palo Alto, CA 94304, USA.

(d_{2m-9}, e_{2m-9}) and (d_{2m-8}, e_{2m-8}) are given by (τ_1, τ_2) and $(-\tau_1, 7)$, respectively, where τ_1 is the smallest odd integer in S which is at least $\sqrt{(h^2/2)+49}$, and τ_2 is the largest odd integer in S which is at most $h/2$. We remove $\{d_i, e_i\}$ from S_{C2} if $d_i < -h$.[3]

- $S_{C3} = \{0, -3, 3, -5, 5, 6, -7, -9, 9, 10, -11, 12, -13, 14\}$
- $S_D = \{\pm 2^i\}_{i=0}^{m-2}$.

We will assume hereafter that h is large enough, in which case the sets S_{C2}, S_{C3}, and S_D are pairwise disjoint.[4] The set S_{C3} has the property that every odd integer r between 1 and 63 can be associated with a balanced vector $\underline{a}_r = (a_{r,w})_{w \in S_{C3}} \in F^{14}$ such that $r = \sum_{w \in S_{C3}} a_{r,w} w^2$. We let S_0 be the union $S_{C2} \cup S_{C3} \cup S_D$. Note that $|S_0| \leq 2(2m-7) + 14 + 2(m-1) = 6m-2$.

For a word $\underline{x}$ of length n and a subset A of S, we will use the notation $\langle \underline{x} \rangle_A$ for the subword of $\underline{x}$ indexed by A.

The algorithm is summarized in Figure 1.

Step A: Initialization of $\underline{x}$

Let $\langle \underline{x} \rangle_{S \setminus S_0} \leftarrow \underline{y}$ and $\langle \underline{x} \rangle_{S_0} \leftarrow \underline{0}$.

Step B: Reduction of $|\sigma_0(\underline{x})|$

For increasing values of indexes $j = -h, -h+1, \ldots$, negate x_j (i.e., let $x_j \leftarrow -x_j$) until $\underline{x}$ becomes balanced. Let j_B be the number of negations performed until this condition is met.

Step C: Reduction of $|\sigma_2(\underline{x})|$

Step C1: Shift cyclically the entries of $\langle \underline{x} \rangle_{S \setminus S_0}$, until the resulting $\underline{x}$ is such that $|\sigma_2(\underline{x})| \leq h^2$. Let j_C be the smallest number of shifts applied until this condition is met.

Step C2: For decreasing values of $i = 2m-8, 2m-9, \ldots, 0$, reduce the value of $|\sigma_2(\underline{x})|$ by assigning $x_{d_i} = -x_{e_i} = -1$ if $\sigma_2(\underline{x}) \geq 0$ and $x_{d_i} = -x_{e_i} = 1$ otherwise.

Step C3: Let $\langle \underline{x} \rangle_{S_{C3}} \leftarrow \underline{a}_{|\sigma_2(\underline{x})|}$; negate $\langle \underline{x} \rangle_{S_{C3}}$ if $\sigma_2(\underline{x}) \geq 0$.

Step D: Reduction of $|\sigma_1(\underline{x})|$

Step D1: For increasing values of indexes $j = 1, 2, \ldots$, swap x_j with x_{-j} until $|\sigma_1(\underline{x})| \leq 2(h-1)$, and let j_D denote the number of swaps made until this condition is met.

Step D2: For decreasing values of $i = m-2, m-3, \ldots, 0$, reduce the value of $|\sigma_1(\underline{x})|$ by assigning $x_{2^i} = -x_{-2^i} = -1$ if $\sigma_1(\underline{x}) \geq 0$ and $x_{2^i} = -x_{-2^i} = 1$ otherwise.

Step E: Recursive encoding

Apply Step A–D recursively to the binary representation of (j_B, j_C, j_D). Concatenate the resulting word with $\underline{x}$ as the final output of the encoder.

Figure 1: Third-order spectral-null encoder.

III. Redundancy Analysis

There is no direct redundancy penalty in Step B, but we need m bits to represent the negation counter j_B which will contribute to the redundancy in Step E. Steps C and D require $|S_0| \leq 6m-2$ bits to reduce $|\sigma_2(\underline{x})|$ and $|\sigma_1(\underline{x})|$ to zero. We also need m bits to represent the shift counter j_C and $m-1$ bits to represent the swap counter j_D. In Step E, the encoding procedure is applied recursively to the $3m-1$ bits that represent (j_B, j_C, j_D), thus generating a word $\underline{x}' \in \mathcal{S}(m', 3)$ of length $m' = 3m + O(\log m)$. Since $m = \lceil \log_2 n \rceil$, it follows that the total redundancy of the encoding scheme is $9 \log_2 n + O(\log \log n)$ bits. This expression will be an upper bound on the redundancy also if we replace n by the overall length, $n + m'$, of the output word.

IV. Time and Space Complexity

Step B can be implemented by first computing the initial value of $\sigma_0(\underline{x})$ and then updating this value for each negation. This requires $O(n)$ increments/decrements of a $\lceil \log_2 n \rceil$-bit counter. Steps C2, C3, and D are rather straightforward and can be implemented using $O(n)$ integer additions. An efficient implementation of Step C1 can be obtained through an indirect method to compute $|\sigma_2(\underline{x})|$ for each shift, using certain look-up tables.

The overall time and space complexity of our encoding algorithm is as follows:

- $O(n)$ additions of $O(\log n)$-bit integers,
- $O(n)$ accesses to $O(1)$ tables, each of size $< n$, and —
- $O(n)$ increments/decrements of $O(\log n)$ counters, each $\lceil \log_2 n \rceil$ bits long.

References

[1] S. Al-Bassam, B. Bose, *On balanced codes, IEEE Trans. Inform. Theory,* IT-36 (1990), 406–408.

[2] R. Karabed, P.H. Siegel, *Matched spectral-null codes for partial-response channels, IEEE Trans. Inform. Theory,* IT-37 (1991), 818–855.

[3] D.E. Knuth, *Efficient balanced codes, IEEE Trans. Inform. Theory,* IT-32 (1986), 51–53.

[4] K.A.S. Immink, *Coding Techniques for Digital Recorders,* London: Prentice-Hall, 1991.

[5] C.M. Monti, G.L. Pierobon, *Codes with a multiple spectral null at zero frequency, IEEE Trans. Inform. Theory,* IT-35 (1989), 463–472.

[6] R.M. Roth, P.H. Siegel, A. Vardy, *High-order spectral-null codes: Constructions and bounds, IEEE Trans. Inform. Theory,* IT-40 (1994), 1826–1840.

[7] L.G. Tallini, S. Al-Bassam, B. Bose, *On efficient high-order spectral-null codes, Proc. IEEE Int'l Symp. Inform. Theory,* Whistler, BC, Canada (Sept. 1995), p. 144.

[3] This can happen only for $i = 2m-10, 2m-11$. Nevertheless, in those cases where only $\{d_{2m-10}, e_{2m-10}\}$ can be removed, then $\{d_{2m-9}, e_{2m-9}\}$ is redundant as well. In fact, it turns out that we will need all the $2(2m-7)$ elements of S_{C2} only when h is close in value to a power of 2.

[4] Some elements in S_{C2} may sometimes be excluded. This allows to have h as small as 18.

Optimal Mappings of the Spectrum of BPSK/QPSK Sequences to Finite Polynomial Fields and Rings

Extended Abstract

M.G.Parker, S.J.Shepherd, S.K.Barton,
Telecommunications Research Group,
Department of Electronic and Electrical Engineering,
University of Bradford,
Bradford, BD7 1DP, UK.
e-mail: mgparker@bradford.ac.uk

Abstract — **It is shown how each bin of the Discrete Fourier Transform (DFT) of a Phase Shift Keyed (PSK) sequence can be optimally mapped to a finite polynomial field or ring. This suggests novel solutions to the VLSI implementation of DFTs, and may also help in the search for spectrally-flat PSK sequences.**

I. Introduction

CONSIDER the N-point DFT of a P-PSK sequence, $d = (d_0, d_1, \ldots, d_{N-1})$, given by,

$$v(n) = \sum_{k=0}^{N-1} d_k e^{j2\pi\frac{kn}{N}} \qquad 0 \le n < N \tag{1}$$

where

$$d_k \in \{1, e^{j\frac{2\pi}{P}}, e^{j\frac{4\pi}{P}}, \ldots, e^{j\frac{(P-1)2\pi}{P}}\}.$$

With $r = \text{lcm}(N, P)$ and $x = e^{j\frac{2\pi}{r}}$, (1) can be expressed as,

$$v_n(x) = \sum_{k=0}^{N-1} d_k(x) x^{\frac{rnk}{N}} \bmod \Phi_r(x) \qquad 0 \le n < N \tag{2}$$

where $\Phi_r(x)$ is the r-th cyclotomic polynomial of degree $\phi(r)$ (ϕ is Euler's Totient Function) [1], and $\deg(v_n(x)) < \phi(r)$. The constellation of polynomials, $\mathbf{V}_n = \{v_n(x)\}$, represent mutually unique points in the complex plane for each bin, n. The 'Pols' column of Tables 1 and 2 shows the constellation size for each bin. Bins, n, which have the same value of $\gcd(N, n)$, generate identical constellations in the complex plane. Therefore only one representative from each class of $\gcd(N, n)$ is tabulated. Note:

The number of polynomials for bin 0 are,

For BPSK: $N + 1$. For QPSK: $(N + 1)^2$.

The number of polynomials for bin 1 when N is prime are,

For BPSK: $2^N - 1$. For QPSK: $(2^N - 1)^2$.

II. Mapping Constellations to Finite Polynomial Fields/Rings

Firstly, the polynomial degree of the constellation representation will be minimised by converting to a polynomial in y. Let $t = \gcd\left(\frac{N}{\gcd(n,N)}, P\right)$. Then substituting $y = x^{\frac{r}{t}}$, $\exists w_n(y)$ such that,

$$w_n(y) \bmod \Phi_t(y) = w_n(x^{\frac{r}{t}}) \bmod \Phi_r(x) = v_n(x) \tag{3}$$

where $\deg(w_n(y)) < \phi(t)$. One can further map the constellation for bin n from the set $\mathbf{W}_n = \{w_n(y)\}$, (or $\mathbf{V}_n = \{v_n(x)\}$) to the field or ring of finite polynomials, $\mathbf{F}_n = \{f_n(u)\}$, mod $M(u)$, mod m, (i.e. the finite polynomial field/ring, $Z_m[u]/M(u)$). One of the conditions for each element of $\mathbf{W}_n$ to map to a unique element of $\mathbf{F}_n$ is,

$$\exists \alpha(u) \in Z_m[u]/M(u), \alpha(u)^t = 1, \alpha(u)^s \ne 1, 0 < s < t \tag{4}$$

To find a suitable $Z_m[u]/M(u)$ that, for a given n, gives a unique mapping from $\mathbf{W}_n$ to $\mathbf{F}_n$ and satisfies (4), the following procedure was adopted.

1. Assign P and N.
2. Assign bin number, n.
3. Compute $\mathbf{V}_n$ using (2), for all d.
4. Re-express $\mathbf{V}_n$ as $\mathbf{W}_n$ using (3).
5. Choose a $Z_m[u]/M(u)$ that satisfies (4). (Ideally $Z_m[u]/M(u)$ should have as few elements as possible but this must be at least equal to the constellation size).
6. Compute
$$\mathbf{F}_n = \{f_n(u) = w_n(\alpha(u)) \bmod M(u) \bmod m\}.$$
7. If there is a one-to-one mapping from $\mathbf{W}_n$ to $\mathbf{F}_n$, then bin n of the N-point DFT of a length N P-PSK sequence can be computed using $Z_m[u]/M(u)$. Go to step 2. Otherwise go to step 5.

Tables 1,2 present finite integer or polynomial mappings for BPSK and QPSK, respectively. The mappings are one-to-one (apart from bin 0 when N is odd), and are therefore optimal. Observe that, for bin 1, N prime, $P = 2$, $\alpha(u)$ must be a root of 2, mod m.

III. Conclusion

The mappings shown suggest efficient hardware solutions for the DFT inherent to OFDM systems [2, 3]. (For example, to compute bins 1 and 2 of a 3-point QPSK DFT:

$$f_n(u) = \sum_{k=0}^{2} d'_k(u)(2u)^{4nk}, \ \bmod\ (u^2 + 1), \ \bmod\ 7,$$

where $d'_k(u) \in \{1, 6u, 6, u\}$ and $2u$ has order 12 over $Z_7[u]/(u^2 + 1)$.) Moreover, the allocation of different mappings for different bin numbers suggests a prime-factor decomposition of the DFT over different finite polynomial fields/rings [1]. Finally, it is hoped these mappings will help to categorise PSK sequences by spectral shape [4].

P	N	n	Pols	m	$M(u)$	t	$\alpha(u)$
2	3	0	4	5	–	2	4
		1	7	7	–	6	3
	4	0	5	5	–	2	4
		1	9	3	u^2+1	4	u
		2	5	5	–	2	4
	5	0	6	7	–	2	6
		1	31	31	–	10	27
	6	0	7	7	–	2	6
		1	19	19	–	6	8
		2	19	19	–	6	8
		3	7	7	–	2	6
	7	0	8	9	–	2	8
		1	127	127	–	14	63
	8	0	9	9	–	2	8
		1	81	3	u^4+1	8	u
		2	25	5	u^2+1	4	u
		4	9	9	–	2	8
	9	0	10	11	–	2	10
		1	$343 = 7^3$	7	u^3+2	18	u
		3	37	37	–	6	11
	10	0	11	11	–	2	10
		1	211	211	–	10	23
		2	211	211	–	10	23
		5	11	11	–	2	10
	11	0	12	13	–	2	12
		1	$2047 = 23.89$	2047	–	22	1983

Table 1: Finite Polynomial Mappings for BPSK DFT Output Bins

P	N	n	Pols	m	$M(u)$	t	$\alpha(u)$
4	2	0	9	3	u^2+1	4	u
		1	9	3	u^2+1	4	u
	3	0	16	17	–	4	4
		1	49	7	u^2+1	12	$2u$
	4	0	25	5	u^2+1	4	u
		1	25	5	u^2+1	4	u
		2	25	5	u^2+1	4	u
	5	0	36	37	–	4	6
		1	$961 = 31^2$	31	u^2+1	20	$2u$
	6	0	49	7	u^2+1	4	u
		1	$361 = 19^2$	19	u^2+1	12	$7u$
		2	361	19	u^2+1	12	$7u$
		3	49	7	u^2+1	4	u
	7	0	64	$65 = 5.13$	–	4	8
		1	$16129 = 127^2$	127	u^2+1	28	$2u$
	8	0	81	$9 = 3^2$	u^2+1	4	u
		1	$625 = 5^4$	5	u^4+1	8	u
		2	81	$9 = 3^2$	u^2+1	4	u
		4	81	$9 = 3^2$	u^2+1	4	u
	9	0	100	101	–	4	10
		1	$117649 = 7^6$	7	u^6+2	36	u
		3	$1369 = 37^2$	37	u^2+1	12	$10u$
	10	0	$121 = 11^2$	11	u^2+1	4	u
		1	$44521 = 211^2$	211	u^2+1	20	$55u$
		2	$44521 = 211^2$	211	u^2+1	20	$55u$
		5	$121 = 11^2$	11	u^2+1	4	u

Table 2: Finite Polynomial Mappings for QPSK DFT Output Bins

REFERENCES

[1] R.E.Blahut, **Fast Algorithms for Digital Signal Processing,** Reading, Addison-Wesley, '85

[2] M.G.Parker, "VLSI Algorithms and Architectures for the Implementation of Number-Theoretic Transforms, Residue and Polynomial Residue Number Systems," *PhD thesis, School of Eng, University of Huddersfield, March '95*

[3] W.Y.Zou,Y.Wu, "COFDM: An Overview", *IEEE Trans on Broadcasting,* Vol 41, No 1, pp 1-8, March '95

[4] M.G.Parker,S.J.Shepherd,S.K.Barton, "Multi-Function Coding for Minimisation of Peak Envelope Power and Error Control in Multitone Modulation Systems", *2nd Annual Conference of the Communications Signal Processing and Coding Programme, Sheffield, 22/23 Jan, '97*

Codes and Ciphers–What's the Difference?

James L. Massey
Signal & Information Processing Laboratory
Swiss Federal Institute of Technology, Zürich, CH–8092 Zürich
E–mail address: massey@isi.ee.ethz.ch

Abstract — **It is argued that the essential difference between codes and ciphers is that a cipher is an ensemble of codes indexed by the secret key. Some speculations are offered for developing an appropriate notion of security within this "ensemble-of-codes" context.**

I. Ciphers as Code Ensembles

CONSIDER a block cipher in which an N-bit plaintext $\mathbf{X}$ is converted to an N-bit ciphertext $\mathbf{Y}$ according to a mapping determined by the K-bit secret key $\mathbf{Z}$, i.e., $\mathbf{Y} = E(\mathbf{X}, \mathbf{Z})$ where $E(.,.)$ is the encrypting function. The cipher must be decryptable and hence one must also have $\mathbf{X} = D(\mathbf{Y}, \mathbf{Z})$ where $D(.,.)$ is the decrypting function. Such a cipher can be viewed as an *ensemble* of rate $R = \frac{1}{2}$ codes in which the codes are indexed by the key $\mathbf{Z}$. The $2N$-tuple $[\mathbf{X} \mid \mathbf{Y}]$ is the codeword. The condition that $\mathbf{X}$ uniquely determine $\mathbf{Y}$, and *vice-versa*, for the codewords within one code is just the condition that the first N components, as well as the last N components, of the codeword be an *information set* for the code.

The functions $E(.,.)$ and $D(.,.)$ must be easy to compute for the cipher to be practical. One generally demands in cryptographic practice that the cipher be "secure" against a *chosen-plaintext attack* when the key $\mathbf{Z}$ has been chosen uniformly at random. The attacker is not privy to the actual choice $\mathbf{z}$ of $\mathbf{Z}$, i.e., he or she does not know which of the 2^K codes in the ensemble is in use, but is able to choose plaintexts $\mathbf{X}$ at will and be told the corresponding ciphertext $\mathbf{Y}$. Equivalently, the attacker chooses the N-bit information portion $\mathbf{X}$ and is then told the N-bit redundant portion $\mathbf{Y}$ of the codeword. In this manner, the attacker accumulates (possibly very many) codewords $[\mathbf{x}_1 \mid \mathbf{y}_1], [\mathbf{x}_2 \mid \mathbf{y}_2], \ldots [\mathbf{x}_L \mid \mathbf{y}_L]$, and then has the task of determining the value $\mathbf{z}$ that was actually chosen for the key or, equivalently, of finding the "redundancy-forming" function $E(., \mathbf{z})$ for the code in use. The cipher is "secure" if the required time and memory for such an attack exceeds feasible limits.

II. What Makes a Cipher Secure?

It is apparent that security of a block cipher demands that the corresponding ensemble of codes possesses something akin to the "pairwise-independence" property needed to show that a code ensemble meets the "random coding bound" for the binary symmetric channel. For the ensemble of codes specified by a cipher, the appropriate *pairwise independence* is that, for every choice $\mathbf{x}_1$ and $\mathbf{x}_2$ of $\mathbf{X}$, the redundant portions $E(\mathbf{x}_1, \mathbf{Z})$ and $E(\mathbf{x}_2, \mathbf{Z})$ of the corresponding codewords be independent and uniformly random when $\mathbf{Z}$ is chosen uniformly at random, cf. Theorem 8 in [1]. As the following argument shows, this is far from enough to guarantee security of the cipher, although it does guarantee the futility of a chosen-plaintext attack in which the attacker is limited to $L = 2$ choices of the plaintext.

Consider the cipher corresponding to the ensemble of *affine codes* in which $\mathbf{Y} = \mathbf{X}\mathrm{A}_{\mathbf{Z}} + \mathbf{b}_{\mathbf{Z}}$ where $\mathrm{A}_{\mathbf{Z}}$ is a nonsingular matrix and $\mathbf{b}_{\mathbf{Z}}$ an N-tuple. When $\mathrm{A}_{\mathbf{Z}}$ and $\mathbf{b}_{\mathbf{Z}}$ are independent and uniformly random, the ensemble has the pairwise-independence property. But an attacker who chooses $\mathbf{x}_i = \mathbf{e}_i$, the i^{th} unit vector, for $i = 1, 2, \ldots N$, is rewarded with $\mathbf{y}_i = \mathbf{r}_i + \mathbf{b}$ where $\mathbf{r}_i$ is the i^{th} row of $\mathrm{A}_{\mathbf{Z}}$ and $\mathbf{b}$ is the value of $\mathbf{b}_{\mathbf{Z}}$. The choice $\mathbf{x}_{N+1} = \mathbf{0}$ returns $\mathbf{y}_{N+1} = \mathbf{b}$ so that the attacker can completely determine the encrypting function with only $L = N + 1$ choices of plaintexts and virtually no computation.

These considerations suggest that some sort of "L-wise independence", for some fairly large L, of the codewords in the ensemble of codes corresponding to a cipher is needed for security. But $E(\mathbf{x}_1, \mathbf{Z}), E(\mathbf{x}_2, \mathbf{Z}), \ldots E(\mathbf{x}_L, \mathbf{Z})$ can all be independent and uniformly random only if the length K of the key $\mathbf{Z}$ is at least LN bits. Such a large key is generally impractical. However, the proper goal of practical cipher design is not to make the cipher impossible to break but rather to make it infeasibly difficult to break. Some kind of "L-wise independence", for some fairly large L, is indeed useful in forcing the attacker to consider many plaintext/ciphertext pairs simultaneously, but one cannot demand that all the resulting equations for the components of the key must also be simultaneously considered. Instead, it appears that one should demand only that fairly "localized" segments of the plaintext/ciphertext pairs be independent, but then also require that the dependence take the form of nonlinear equations that are not readily solved.

Certain local properties of functions that may be useful in developing an appropriate notion of L-wise local independence have already been considered for the building blocks within ciphers. If f is a mapping from n bits to k bits and $\mathbf{R}$ is a uniformly random n-tuple, one says that f is m^{th}-order *correlation immune* [2] if $f(\mathbf{R})$ is independent of every m-bit subvector of $\mathbf{R}$. One says that f is *t-resilient*, cf. [3], if, when conditioned on any t-bit subvector of $\mathbf{R}$, $f(\mathbf{R})$ is uniformly random. One says that f is *locally randomizing* [4] of order r if every r-bit subvector of $f(\mathbf{R})$ is uniformly random. These notions of locality are obviously interrelated and a coding theorist will recognize that they have to do with the *dual distance* of an appropriately defined (in general nonlinear) code whose number of codewords is a power of 2, i.e., with the length of the shortest subword that is not uniformly random when a codeword is chosen uniformly at random.

References

[1] J. L. Massey, *Threshold Decoding.* Cambridge, MA: M.I.T. Press, 1963.

[2] T. Siegenthaler, "Correlation-Immunity of Nonlinear Combining Functions for Cryptographic Applications," *IEEE Trans. Info. Th.*, vol. IT-30, pp. 776-780, Oct. 1984.

[3] D. R. Stinson and J. L. Massey, "An Infinite Class of Counterexamples to a Conjecture Concerning Non-linear Resilient Functions," *J. Cryptology*, Vol. 8, No. 3, pp. 167-173, 1995.

[4] U. M. Maurer and J. L. Massey, "Local Randomness in Pseudorandom Sequences," *J. Cryptology*, Vol. 4, No. 2, pp. 135- 149, 1991.

Hard Problems from Coding Theory and their Applications in Cryptography

Jacques Stern
Laboratoire d'informatique,
École Normale Supérieure

Abstract — **The present paper briefly reviews several cryptographic schemes that have been proposed in the literature and whose common feature is to use error-correcting codes. These schemes cover various aspects of cryptography such as public key cryptosystems, zero knowledge identification or pseudorandom number generation. Before the review, an attempt is made to put the hardness assumptions on which they rely in a proper complexity-theoretic perspective.**

I. Hard Problems from Coding Theory

We assume familiarity with the basic notions related to error-correcting codes and we restrict our attention to linear binary codes. As usual, we denote by $\omega_H(x)$ the Hamming weight of a vector x. It is well known that, even in this simple case, the question of finding the closest codeword to a vector is hard. It is also difficult to find a word of given weight from its syndrome's value (see [2]). More precisely, the following problem, stated in the style of [10], is NP-complete:

Instance An $m \times n$ binary matrix $H = (h_{ij})$, a binary non-zero vector $y = (y_1, \ldots, y_m)$, and a positive integer w.

Question Is there a binary vector $x = (x_1, \ldots, x_n)$ with $\omega_H(x) \leq w$ such that, for $1 \leq i \leq m$, $\sum_{j=1}^{n} h_{ij} \cdot x_j \equiv y_i \pmod 2$?

Comment: The variant in which $y = (0, 0, \ldots, 0)$ has remained an open problem for a long time and has only recently been proved NP-complete by A. Vardy (see [26]).

NP-completeness ensures that there is no polynomial-time algorithm for solving a problem in the worst case; however many NP-complete problems can be attacked after a suitable preprocessing phase or can be suitably approximated. This is not the case for the above problem as shown in the literature (see e.g.[4, 1]). As for the hardness of random instances, especially for families of random codes with a constant information rate, it is known that they almost surely satisfy the Gilbert-Varshamov bound ([17]) and therefore that they can correct a constant fraction of the length of the codewords. Still, it is difficult do decode with respect to these codes and it even looks equally hard to exhibit codewords achieving the minimum distance or to find words of given syndrome whose weight is close to the minimum distance. Several probabilistic algorithms have been proposed that solve these problems (see [15, 23, 5]) but their running time is exponential.

In order to encapture the notion of hardness that was just described, we make the following definition, where $\mathcal{M}(p \times q)$ denotes the set of binary matrices with p rows and q columns:

Let θ, δ be in $(0,1)$; The $SD(\theta, \delta)$ collection is the set of functions $\{f_n\}$ such that:

$$D_n = \{(M, x) | M \in \mathcal{M}(\lfloor \theta n \rfloor \times n), x \in \{0,1\}^n, \omega_H(x) = \lfloor \delta n \rfloor\}$$
$$f_n : \quad (M, x) \quad \longmapsto \quad (M, M \cdot x)$$

With this definition, we can formally state an intractability assumption.

Intractability assumption *Let θ be in $(0,1)$. Then, for all δ such that $0 < \delta < 1/2$ and $H_2(\delta) < \theta$, the $SD(\theta, \delta)$ collection of functions is strongly one-way.*

Strongly one-way functions are standard objects of complexity theory: the practical meaning of the above is that it is virtually unfeasible to compute a word of given syndrome whose weight is slightly below the minimum distance. Note that our hypothesis, albeit plausible, is formally stronger than the usual statement that it is difficult to decode under half the minimum distance. This statement only covers the more restrictive case where $H_2(2\delta) < \theta$ holds.

Of course, the above intractability assumption does not tell much about the choice of actual parameters that guarantee that the concrete problem of finding short codewords is beyond the limits of current computing technology. A survey of known algorithms for solving this problem appears in [5] with a discussion of their possible implementations and of their actual performances. We refer the reader to this paper and we only briefly comment on some figures taken from [5], for the case $\theta = 1/2$. When $n = 256$ and $m = 128$ finding a word of weight close to 30 from its syndrome is possible and takes a few hours on a workstation. On the other hand, if we set $n = 512$, $m = 256$ and look for a word of weight 56, the workfactor for the search based on the best algorithm can be estimated as 2^{70}. This would involve a major computing effort. If one is only interested into decoding under half the minimum distance, one should go to much larger dimensions such as $n = 1024$, $m = 512$ to obtain a similar security level.

II. Public key cryptography

Public key cryptography goes back to the seminal work of Diffie and Hellman (see [6]), dated 1976. Shortly afterwards, the celebrated RSA algorithm was discovered and published in [19]. About at the same time, R. Mc Eliece proposed in [18] a public key cryptosystem based on coding theory. The idea was to start with a structured code with an efficient decoding algorithm and to hide the structure by secret transformations.

The secret key of the system consists of a $k \times n$ generating matrix G for a Goppa code which can correct t errors, of a nonsingular $k \times k$ matrix S and of an $n \times n$ permutation matrix P. The public key is the "scrambled" matrix $\tilde{G} = SGP$, which appears as the generating matrix of an unstructured code.

Plaintexts are strings of k bits. They are encoded by randomly choosing an error vector e of dimension n with weigth at most t; the ciphertext reads $c = m\tilde{G} + e$. To decrypt, one successively computes $c' = cP^{-1}$, decodes c' w.r.t. G which yields m' and recovers m as $m'S^{-1}$.

In [18], it is suggested to use $n = 1024$, $k = 524$ and $t = 50$. This is consistent with the discussion of paragraph I. No successful cryptanalysis has been reported against the Mc Eliece cryptosystem. However, when compared with RSA, it suffers from two drawbacks: the size of keys and the data expansion produced by encryption.

III. Zero-knowledge Identification

Zero-knowledge proofs were introduced in 1985, in a paper by Goldwasser, Micali and Rackoff ([13]). Their practical significance for public key identification was soon demonstrated in the work of Fiat and Shamir ([8]). Zero-knowledge identification allows a user to convince another entity of his identity by means of an on-line communication without giving enough information to allow anyone else to misrepresent himself as the legitimate user, including the entity carrying the identification process.

In [24, 25] we proposed a new identification scheme, based on the hardness of the SD problem dicussed in section I. The proposed scheme uses a fixed $(m \times n)$-matrix H over the two-element field and, for each user U, a secret key s_U, chosen among all n-bit words with a prescribed number p of 1's. The public identification of the user is computed as $i_U = H.s_U$ Any user U (the "prover") identifies himself to a "verifier" by performing repeatedly the following protocol where$\langle\rangle$ denotes some public cryptographic hash function such that MD5 (see [20]) or SHA (see [21]) and where other notations are standard:

1. The prover picks a random n-bit word y together with a random permutation σ of the integers $\{1 \cdots n\}$ and sends c_1, c_2, c_3 respectively as

$$c_1 = \langle \sigma || H(y) \rangle$$
$$c_2 = \langle y.\sigma \rangle$$
$$c_3 = \langle (y \oplus s_U).\sigma \rangle$$

   to the verifier.

2. The verifier sends a random element b of $\{0, 1, 2\}$.
3. If b is 0, then, the prover returns y and σ. If b is 1 then, the prover reveals $y \oplus s$ and σ. Finally, if b equals 2, then the prover discloses both $y.\sigma$ and $s_U.\sigma$.
4. If b equals 0, the verifier checks that c_1 and c_2, which were sent in step 1, have been computed honestly.

   If b equals 1, the verifier checks that c_1 and c_3 were correct.

   Finally, if b is 2, the verifier checks the weight property and correctness of c_2 and c_3.

Possible sizes are as follows :

- $n = 512, m = 256, p = 56$
- $n = 768, m = 384, p = 84$
- $n = 1024, m = 512, p = 110$

They lead to a scheme that be implemented on a standard smart card with no arithmetical co-processor. An implementation has been done on an SGS Thomson ST16623 card with 224 bytes of RAM, of which 140 bytes only are used for SD. One round is computed by the card in 800 ms.

IV. Pseudorandom Number Generation

A pseudorandom generator is an algorithm producing strings of bits that look random. The concept of "randomly looking" has been formalized by Blum and Micali [3] within the framework of complexity theory. Subsequently, a long series of deep articles led to the conclusion that the existence of a one-way function is equivalent to the hypothesis that a pseudorandom generator exists [14]. In more practical terms, this has produced a notion of "proven security", i.e. based on the difficulty of well known problems like factorization. In an early work [12], Goldreich, Krawczyk and Luby established that the existence of a pseudorandom generator could be based on hard problems from the theory of error-correcting codes. Unfortunately, their construction was a bit intricate. In [9], we presented a new scheme based on the syndrome decoding problem. Our scheme is extremely simple and achieves quadratic time with respect to the security parameters for producing a linear amount of random bits.

We construct the generator G form parameters θ, δ in the following way, where notations come from section I:

Input: $(M, x) \in D_n$

Output: $(M, M \cdot x)$

Following a standard construction, we also consider an iterative generator g that, on input (M, x), outputs as many bits as we like. To perform this iteration, we use an efficient algorithm that computes a vector of size n and weight $\lfloor \delta n \rfloor$ from a number of the appropriate bitsize ρn. Th iterative generator goes as follows:

```
Input: $(M, x) \in D_n$
  1. compute $y = M \cdot x$
  2. separate $y$ in two bit strings $y_1$ and $y_2$, $y_1$ being the first
     $\rho n$ bits of $y$ and $y_2$ the remaining bits.
  3. output $y_2$
  4. set $x \leftarrow A(y_1)$ and goto 1.
```

References

[1] S. Arora, L. Babai, J. Stern and Z. Sweedyk. The hardness of approximate optima in lattices, codes and systems of linear equations, *Proc. 34th Ann. Symp. on Foundations of Computer Science*, (1993), 724–733.

[2] E. R. Berlekamp, R. J. Mc Eliece and H. C. A. Van Tilborg. On the inherent intractability of certain coding problems, *IEEE Trans. Inform. Theory*, (1978) 384–386.

[3] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM J. Computing*, 13(4):850–863, 1984.

[4] J. Bruck and M. Naor. The hardness of decoding linear codes with preprocessing, *IEEE Trans. Inform. Theory*, IT-36(2) (1980), 381–385.

[5] F. Chabaud. On the security of some cryptosystems based on error correcting codes. *Proceedings of Eurocrypt 94*, Lecture Notes in Computer Science 950, 131–139.

[6] W. Diffie and M. E. Hellman. New Directions in Cryptography, *IEEE Trans. Inform. Theory*, IT-22, (1976), 644–654.

[7] U. Feige, A. Fiat and A. Shamir, Zero-knowledge proofs of identity, *Proc. 19th ACM Symp. Theory of Computing*, (1987), 210–217, and *J. Cryptology*, **1** (1988), 77–95.

[8] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems, *Proceedings of Crypto 86*, Lecture Notes in Computer Science 263, 181–187.

[9] J.-B. Fischer & J. Stern. A pseudo-random generator provably as secure as syndrome decoding, *Proceedings of Eurocrypt 96*, Lecture Notes in Comp Sci 1070 (1996), pp. 245–255.

[10] M. R. Garey and D. S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness.* W. H. Freeman and Co, 1979.

[11] O. Goldreich. *Foundations of cryptography (Fragments of a book).* Weizmann Institut of Science, 1995.

[12] O. Goldreich, H. Krawczyk, and M. Luby. On the existence of pseudorandom generators. *Proc. 29th Symp. on Foundations of Computing Science*, pages 12–24. IEEE, 1988.

[13] S. Goldwasser, S. Micali and C. Rackoff. The knowledge complexity of interactive proof systems, *Proc. 17th ACM Symp. Theory of Computing*, (1985), 291–304.

[14] R. Impagliazzo, L. Levin and M. Luby. Pseudorandom generation form one-way functions, *Proc. 21st ACM Symp. Theory of Computing*, (1989), 12–24.

[15] J . S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Trans. Inform. Theory*, IT-34(5): 1354–1359.

[16] F. J. MacWilliams and N. J. A. Sloane. The theory of error-correcting codes, North-Holland, Amsterdam-New-York-Oxford (1977).

[17] J. N. Pierce. Limit distributions of the minimum distance of random linear codes, *IEEE Trans. Inform. Theory*, (1967), 595–599.

[18] R. J. McEliece. A Public-Key System Based on Algebraic Coding Theory, Jet Propulsion Lab, *DSN Progress Report 44*, (1978), 114–116.

[19] R. Rivest, A. Shamir and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems, *Comm. of the ACM* 21-2 (1978), 120–126.

[20] R. L. Rivest. The MD5 Message Digest Algorithm. *Proceedings of Crypto 90*, Lecture Notes in Computer Science 537, 303–311.

[21] U. S. Department of Commerce, National Institute of Standards and Technology. Secure Hash Standard. Federal Information Processing Standard Publication 180, 1993.

[22] J. Stern. An alternative to the Fiat-Shamir protocol, *Proceedings of Eurocrypt 89*, Lecture Notes in Computer Science 434, 173–180.

[23] J. Stern. A method for finding codewords of small weight, *Coding Theory and Applications*, Lecture Notes in Computer Science 388, 106–113.

[24] J. Stern. A new identification scheme based on syndrome decoding. *Proceedings of Crypto 93*, Lecture Notes in Computer Science 773, 13–21.

[25] J. Stern A new paradigm for public key identification, *IEEE Transactions on Information Theory*, 42 (6) 1996, pp. 1757–1768.

[26] A. Vardy, The intractability of computing the minimum distance of a code, manuscript dated Nov. 5, 1996.

Coding problems in cryptography or codesets vs codewords

G.R. Blakley‡, T.Johansson§, G.Kabatianski†[1]

†Institute for Problems of Information Transmission,
Bolshoi Karetniy, 19, Moscow, 101447, Russia; Email: kaba@ippi.ac.msk.su
‡ Dept. of Math., Texas A&M University, College Station, TX 77843-3368, USA
§Dept. of Information Tech., Lund University,S-22100 Lund, Sweden.

Abstract — **We consider relationships between coding theory and the following cryptographic problems : the wire-tap channel II, secret sharing, unconditional authentication, and hash functions.**

I. Introduction

CONTRARY to a widely shared opinion, there are not so many well-established links between coding theory and cryptography. In this lecture we consider the following problems, which originated in cryptography but were later recognized and/or treated as coding-theoretic problems: Wire-Tap Channel, Secret Sharing, Unconditional Authentication, Hash Functions. Our goal is to revise "old bridges" between these subjects and coding theory and to construct new ones.

II. Wire-tap channel II

SUPPOSE a set of M messages is transmitted by n-bit vectors over a noiseless channel and there is an intruder who can observe a subset of coordinates of size m at his choice. The sender wants to provide a correct recovery of a message at the receiving end and to minimize the information that the intruder can gain from the eavesdropped bits (wire-tap channel II [1]). To do this the sender encodes messages as subsets of codewords rather than as single words, and to send a message he chooses randomly and sends one of words (vectors) from the corresponding subset. Though such a situation, in which a subset of the set of all words rather than a single word, is used to send a message is not common for coding theory, is also not entirely new. It occurs, for example, in codes correcting "defects" [2], and codes correcting localized errors [3] (note that in [2], [3] the usage subsets instead of single words is motivated by the sender's knowledge of some *a priory* information about the channel, which is not known to the receiver). The most natural choice of "codesets" is a linear (n, k)-code V and its cosets . This choice has been investigated in [1] from the probabilistic point of view. Later a more combinatorial approach based on the notion of generalized Hamming weights (GHW) was introduced in [4], what was in fact the second birth of the early introduced notion of minimal support weights [5]. It is easy to see that the intruder has no information from any m observed positions iff $m < d(V^{\perp})$, where $d(U)$ is the minimum Hamming distance of a code U and $V^{\perp}$ denotes the code dual to the code V. More generally, the intruder can get exactly $m - dimV_I$ bits of information from the observed set I of positions, where $I = \{i_1, \dots, i_m\}$ and V_I is a projection of the code V on positions in the set I. On the other hand, $m - dimV_I = dim(V^{\perp})^I$ (since from linear algebra $V_I^{\perp} = (V^{\perp})^I)$, where $U^I = \{(u_{i_1}, \dots, u_{i_m}) : u \in U, u_j = 0$ for $j \notin I\}$. Hence the intruder cannot get more then $j - 1$ bits of information if he observes less then $d_j(V^{\perp})$ positions, where $d_j(U) = min\{|I| : dimU^I = j\}$ is called j-th minimal support weight (MSW)[5] or j-th GHW [4]. There are a few generalizations of this notion to the general case (nonlinear codes). The definition in [6] is based on the following useful notion. Denote by $supp(V)$ the set of coordinates in which not all words of A are equal. Then define the function of supports $S(l) = \min\{|supp(A)| : A \subseteq V, |A| = l\}$ and define j-th MSW of the code V as the j-th value of the function $S(x)$. In Section 5 we give one more reason in support this definition of support weights.

III. Secret Sharing Schemes

INFORMALLY speaking, an (n, k)-threshold secret sharing scheme (SSS) makes it possible to share a secret among n participants in such a way that any sets of k or more participants can recover the secret exactly, while fewer participants can get no additional (*i.e. a posteriori*) information about the possible value of the secret (the SSS with this last property is called *perfect*). Formally, there is a set $\mathcal{S}_0$ of all possible secrets from which a secret s_0 is chosen with probability $p(s_0)$. And there is a dealer of the SSS who "shares" the secret s_0 by distributing shares $s_1, \dots, s_n$ chosen with probability $P_{s_0}(s_1, \dots, s_n)$. Define the probability distribution P on a set $\mathcal{S} = \mathcal{S}_0 \times \dots \times \mathcal{S}_n$, where $P(s) = P(s_0, s_1, \dots, s_n) = p(s_0)P_{s_0}(s_1, \dots, s_n)$. A set $V = \{s \in \mathcal{S} \mid P(s) > 0\}$ is called the "code" of the SSS $(P, \mathcal{S})$. There are some relationships between codes and SSS. For instance, *ideal* (*i.e.*, $|\mathcal{S}_0| = |\mathcal{S}_i|$ or $H(\mathcal{S}_0) = H(\mathcal{S}_i)$) (n, k)-threshold SSS is perfect iff its code is (n+1,k)-MDS code. Now we introduce another notion of a code of SSS, in a way similar to the wire-tap channel. Consider a code of length n consisting of $|\mathcal{S}_0|$ messages (secrets), where the "codeset" $\{(s_1, \dots, s_n) : P_{s_0}(s_1, \dots, s_n) > 0$ corresponds to a message s_0. Then the (virtual) intruder can get no a posteriori information after observing $k-1$ or less positions, but we demand (in addition to the wire-tap model) that he can recover the message after observing any k positions.

After the original papers [7], [8] many other extensions of the statement of the problem have been considered. Among them, SSS with *veto* capability. In such schemes, any *veto* coalition of t or more participants can (by sending spurious shares)) bar any other coalition from learning the secret. Nevertheless, any k or more participants can recover the secret if the number of *veto* participants is less than t. The notion of the "code" of a SSS introduced above helps to show [9] (in contrast to earlier papers) that ideal perfect $(n, k; t)$-threshold schemes with veto

[1] Part of this work was supported by the Russian Fundamental Research Foundation project 96-01-00884 and by a Visiting Fellowship grant of the Royal Swedish Academy of Sciences.

capability exist only if $t = 1$ and $k = n - 1$. On the other hand, in [9] we construct ideal perfect $(q, q-1; 1)$-threshold schemes with veto capability, where q is a power of a prime number. Various researches considered the problem of SSS-s with cheaters. We want to introduce and discuss a new notion of coding theory, namely, error-localizing codes, which is motivated by SSS-s capable of identifying cheaters (see [10]). We say that a code V can localize t errors if $| \cup supp\{v, y\}| \leq t$ for any y, where $v \in V : d(v, y) \leq t$, *i.e.*, for any (received) vector y there is a set $L \subset \{1, \ldots, n\} : |L| = t$ such that any code vector $v \in V : d(v, y) \leq t$ may differ from y only in positions of the set L. In fact, this notion is also motivated by the usual procedure of decoding of nonbinary codes, namely, first, finding positions (locators) of errors, and second, finding error values. It is easy to see that a "good" code localizing t errors automatically produces a "good" code correcting t errors, whose complexity of decoding is determined by the complexity of error-localizing decoding. This notion can be generalized in the following way. We say that a code V localizes t errors by T-coverings if for any y there exists a set $C = C(y, V) \in \{1, \ldots, n\}$ such that $|C| = T$ and $supp\{v, y\} \subseteq C$ for all $v \in V : d(v, y) \leq t$. Denote by $r_l(n, t, T)$ the minimal redudancy of a code localizing t errors by T-coverings and by $r(n, d)$ the minimal redudancy of a code with the minimal Hamming distance d. Then

$$r(n, 2t+1) \leq r_l(n, t, T) + r(n, T+1)$$

(localize errors and then correct them as T erasures). In particular, $r_l(n, t) \leq r(n, 2t+1) - r(n, t+1)$, and codes attaining this bound have a nice property that one can split a decoding procedure in the two described above steps and lose nothing by this splitting (therefore existence of such codes should be a very rare event, if they exists at all).

IV. Unconditional authentication and strongly universal hash functions

LET us return to the wire-tap channel. Suppose now that the intruder can observe any transmitted information; suppose that he even knows the original message and tries to replace it by a false one. To provide protection against this the sender uses not a single code but a set of codes and chooses a code randomly. Hence again we have a situation when a subset of the set of all words rather than a single word is used to send a message. As usual in cryptography we assume that the sender's choice is known to the receiver but not to the intruder. For simplicity consider only systematic codes. Let A be the set of messages, F be a set of encoding maps and suppose a transmitted word has the form $(m; z)$, where $m \in A$ is a message, $z = f(m)$ is a "tag" ("parity-check symbols" - in coding theory) from a set B. We may enumerate the encoding maps of $F = \{f_1, \ldots, f_n\}$ and define an A(uthentication)-code which is a $|B|$-ary code V of length n consisting of $|A|$ words $(f_1(m), \ldots, f_n(m)) : m \in A$. Define A-distance (which is not a metric) between x and y as $d_A(x, y) = n - q\gamma(x, y)$, where $q = |B|$, $\gamma(x, y) = \max\{|\{i : x_i = b, y_i = b'\}| : b, b' \in B\}$. Almost w.l.o.g. we assume that all words of the A-code V have the uniform composition, i.e., $|\{i : v_i = b\}| = n/q$ for any $v \in V$, any $b \in B$. Then the maximal probability P_S of success of the intruder's substitution equals $1 - n^{-1}d_A(V)$, where the minimum A-distance of a code is defined as usual. Note that $d_A(V) \leq d(V)$, which allows to get upper bounds on the cardinality of systematic A-codes (see [11]). On the other hand, if C is a linear code containing vector **1**, then its minimal Hamming distance and the minimal A-distance of the factor-code $C/\{\lambda 1\}$ coincide. It gives a way of constructing "good" A-codes [11].

Now we want to point out the relationship between A-codes and almost strongly universal$_2$ hash functions [12]. Informally speaking, hash functions are functions which map a set A into a set B (usually $|A| >> |B|$) and do it like a random mapping. A single function cannot behave as a random one, therefore the problem is to construct a "small" family F of functions such that their "common behavior" has some properties of a random mapping. One of these desired properties is that for any s different elements $x_1, \ldots, x_s \in A$ and any s elements $y_1, \ldots, y_s \in B$ the number of functions such that $f(x_i) = y_i : i = 1, \ldots, s$ is the same, i.e., equals to $|F|/|B|^s$. Such a family is called strongly universal$_s$ [12]. Consider functions as rows of an array (with $|A|$ columns). Then a family of strongly universal$_s$ functions corresponds to an orthogonal $|B|$-ary array of strength s and vice versa.

Consider more general class of functions for the case $s = 2$. A family F is called ϵ-almost strongly universal$_2$ [12] if $|\{f : f(x_1) = y_1\}| = |F|/|B|$ and $|\{f : f(x_1) = y_1, f(x_2) = y_2\}| \leq \epsilon|F|/|B|$. Consider the given family F of ϵ-almost strongly universal$_2$ hash functions as the set of encoding maps. Then $P_S \leq \epsilon$ and the transposed array of this family is the same as A-code with $d_A \geq n(1 - \epsilon)$.

We finish this section with a remark that the described above notions of A-codes and almost strongly universal$_2$ hash functions are very close to codes for identification and pairwise separated measures (see [13]).

V. Perfect hash functions

A set F of functions from $\{1, \ldots, M\}$ to $\{1, \ldots, q\}$ is called a (M, q, s)-perfect (or s-perfect) set of n hash functions if for every $S \subset \{1, \ldots, M\}$, $|S| = s$ there exists $f \in F$ such that $f(x) \neq f(y)$ for all $x, y \in S, x \neq y$. To make a link to coding theory let us generalize this notion.

Define the *s-th hash distance* of a q-ary code V as the maximal number $d_s = d_s(V)$ such that for any s different codewords there are at least d_s coordinates every one of which takes distinct values for these codewords. Note that $d_2(V)$ coincides with the usual minimal Hamming distance of the code V and $d_s(V) = 0$ for $s > q$.

Let us enumerate the codewords of V: $v_1, \ldots, v_M$ and consider the n coordinates as functions from $\{1, \ldots, M\}$ to $\{1, \ldots, q\}$. Then a code of length n with d_s corresponds to an (M, q, s)-perfect set of n hash functions such that for every $S \subset \{1, \ldots, M\}$, $|S| = s$ there are at least d_s functions $f \in F$ such that $f(x) \neq f(y)$ for all $x, y \in S, x \neq y$. In particular, an (M, q, s)-perfect set of n hash functions is the same as a q-ary code of length n and cardinality M with $d_s \geq 1$. It is not difficult to prove that the following recursion is valid for linear codes

$$d_{s-1}(V) - d_s(V) + 1 \geq dim(V),$$

and if we replace $s-1$ by $s-2$, then the corresponding recursion is valid for general (unrestricted) codes. This shows that the notion of hash distance is natural because any set of s-perfect hash functions is automatically a hash code with rather large d_{s-2} (or even d_{s-1} - for linear codes). It also helps to prove a new upper bound on the cardinality of hash codes, which in the particular case of perfect hash function improves the well known Friedman-Komlosh bound [14]. For more details see [15].

Most coding-theoretic problems deal with pairwise characteristics such as minimal Hamming distance between codewords.

Perfect hash functions give an important example of a problem which is based on studying characteristics of s-subsets of a code. This problem and some others can be represent in the following way. Let G be a (sub)set of s-dimensional q-ary vectors, and B be its complement. Define the *s-th G-distance* of a q-ary code V as the maximal number $d_s = d_s(V)$ such that for any given s different codewords there are at least d_s coordinates such that the corresponding s-dimensional vectors belong to the set G. The case of G consisting of all vectors with different coordinates gives the notion of s-th hash distance considered above. The case of B consisting of all constant vectors gives the mentioned in Section 2 definition of MSW-GHW for nonlinear codes (see [6]). Another choice of G will give the definition of so called separated systems (see[16]).

References

[1] L.H.Ozarov and A.D.Winer, "Wire-tap channel II," *AT&T Bell Labs Tech. J.* , vol. 63, pp. 2135-2157, 1984.

[2] A.V. Kuznetsov and B.S. Tsybakov, "Coding for memory with defect cells," *Problems of Information Transmission*, vol. 10, no.2, pp. 52–60, 1974.

[3] L.A. Bassalygo, S.I. Gelfand, and M.S. Pinsker, "Coding for channels with localized errors," in *Proc. 4th Soviet-Swedish Workshop in Information Theory*, pp. 95–99, Gotland, Sweden, 1989.

[4] V. K. Wei , "Generalized Hamming weights for linear codes," *IEEE Transactions on Information Theory*, vol. 37,no. 5, pp. 1412–1418, 1991.

[5] T.Helleseth, T. Klöve, and J.Mykkeltveit, "The weight distribution of irreducible cyclic codes with block lengths $n_1((q^l-)/N)$," *Discr. Math.*, , vol. 18, pp. 179-211, 1977.

[6] L.A.Bassalygo, "Supports of a code," *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, Lect. Notes in Comp. Sci.*, vol. 948, pp. 1–3, 1995.

[7] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of AFIPS 1979 National Computer Conference,* vol. 48, N. Y., pp. 313–317, 1979.

[8] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no.1, pp. 612–613, 1979.

[9] G.R.Blakley and G.A. Kabatianskii, "When perfect secret sharing schemes with veto exist," *private communication*, 1997.

[10] K. Kurosawa, S. Obana, and W. Ogata, "t-cheater identifiable (k,n) threshold secret sharing schemes," *Lect. Notes in Comp. Sci.*, vol. 963, pp. 410–423, 1995.

[11] G.A. Kabatianskii, B. Smeets, and T. Johansson, "On the cardinality of systematic authentication codes via error-correcting codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 566–578, 1996.

[12] M.N. Wegman and J.L. Carter, "New hash functions and their use in authentication and set equality," *J. Comput. Sys. Sci.*, vol. 22, pp. 265–279, 1981.

[13] L.A. Bassalygo, M.V. Burnashev, "Authentication, identification and pairwise separated measures," *Problems of Information Transmission*, vol. 32, no.1, pp. 41–47, 1996.

[14] M.L. Friedman and J. Komlos, "On the size of separating systems and families of perfect hash functions," *SIAM J. Alg. Disc. Methods*, vol. 5, pp. 538–544, 1984.

[15] L.A. Bassalygo, M. Burmester, A.Dyachkov, G. Kabatianski, "Hash codes," in *Proc. of 1997 IEEE Int. Symp. on Information Theory*, Ulm, Germany, 1997.

[16] Y.L. Sagalovich, "Separating systems," *Problems of Information Transmission*, vol. 30, no.2, pp. 14–35, 1996.

Secure Multi-Party Computation and Coding Theory [1]

Ueli Maurer

Department of Computer Science, ETH Zurich, CH-8092 Zurich, Switzerland,
Email: maurer@inf.ethz.ch

Abstract — **We give an overview of recent research in secure multi-party computation, in particular results in the information-theoretic (as opposed to computational) model, and present new results that characterize completely which adversary collusions can be tolerated in an information-theoretically secure multi-party computation.**

I. Introduction

CONSIDER a set of players who do not trust each other. Nevertheless theywant to compute some agreed function of their inputs in a secure way. A classical example is voting in which the sum of the voters' votes is to be computed, assuming here that no trusted authority is available. Security in this context means guaranteeing the correctness of the output of the computation while keeping the players' inputs private[2]. This is the well-known secure multi-party computation problem (e.g. [6, 4]). Applications of secure multi-party computation include various forms of cooperation in the absence of a mutually trusted party.

The protocol must be robust in the sense of tolerating that some players misbehave collectively without breaching the correctness of the result or the privacy of the inputs. Two types of misbehavior are usually considered. In the passive model, collusions of players follow the protocol correctly but pool their complete information in order to violate the other players' privacy. In the active model, a set of coordinated adverse players deviates arbitrarily from the protocol in order to violate the correctness of the result and/or the other players' privacy.

Solutions to the secure multi-party computation problem can be classified according to a number of criteria that are briefly discussed below. Some papers (e.g. [4, 1, 2]) describe protocol constructors which for *any* function generate a protocol for securely computing it, while other approaches are tailored to a particular function like voting (e.g. [3]). The major reason for considering special functions is the potential gain of efficiency compared to a general solution. The communication models differ with respect to whether or not broadcast channels and/or secure communication channels are available, and whether the communication channels are synchronous or asynchronous. Adversaries are classified according to their computational resources (limited, hence cryptographic security, e.g. [4], or unlimited, hence unconditional or information theoretic security, e.g. [1, 2]), and according to whether they cheat actively or passively.

II. Outline of the Talk

In this talk we consider perfect multi-party computation for any function in which correctness and privacy are guaranteed even when the adversaries have infinite computing power. We assume that secure bilateral communication between any pair of players is possible but do not assume broadcast channels. All previous results in the literature specify the sets of adverse players (passive or active) that can be tolerated by their cardinality, i.e. by a threshold. In this setting, Ben-Or, Goldwasser and Wigderson [1] and independently Chaum, Crépeau and Damgård [2] proved that with n players all passive collusions with less than $n/2$ members or, alternatively, all active adversaries with less than $n/3$ members can be tolerated.

These results are based on techniques and results from coding theory and these relations to coding are reviewed in this talk. Furthermore, we show that the above mentioned threshold-type results can be generalized to arbitrary characterizations of potential adversaries. The necessary and sufficient conditions for the existence of secure multi-party protocols in terms of the potentially misbehaving player sets are given. In the passive model, for every function there exists a protocol secure against a set of potential collusions if and only if no two of these collusions add up to the full player set P. In the active model, for every function there exists a protocol secure against a set of potential adverse player sets if and only if no three of these add up to the full player set P.

III. Acknowledgement

The results on general adversary structures are joint work with Martin Hirt [5].

References

[1] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. 20th ACM Symposium on the Theory of Computing (STOC)*, pages 1–10, 1988.

[2] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *Proc. 20th ACM Symposium on the Theory of Computing (STOC)*, pages 11–19, 1988.

[3] R. Cramer, M. Franklin, B. Schoenmakers, and M. Yung. Multi-authority secret-ballot elections with linear work. In *Advances in Cryptology — EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 72–83. IACR, Springer-Verlag, 1996.

[4] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game — a completeness theorem for protocols with honest majority. In *Proc. 19th ACM Symposium on the Theory of Computing (STOC)*, pages 218–229, 1987.

[5] M. Hirt and U. Maurer, Complete characterization of adversaries tolerable in secure multi-party computation, to appear in *Proc. of 16th ACM Symposium on Principles of Distributed Computing (PODC)*, Santa Barbara, Aug. 1997.

[6] Andrew C. Yao. Protocols for secure computations. In *Proc. 23rd IEEE Symposium on the Foundations of Computer Science (FOCS)*, pages 160–164. IEEE, 1982.

[1] This work was supported in part by the Swiss National Science Foundation, grant no. SPP ICS 5003-045293.

[2] More generally, inputs can be provided by parties outside of the player set. Furthermore, a different function can be assigned to each player or party. Each party learns only the corresponding function value. In other words, the players collectively simulate a trusted party computing the function(s) for those providing the inputs.

A PRIVATE KEY CRYPTOSYSTEM WITH PRODUCT CODES

Angela I. Barbero and Juan G. Tena

The first author is with the Dept. of Mathematics Applied to Engineering of the University of Valladolid. E-mail: angbar@wmatem.eis.uva.es. The second author is with the Dept. of Algebra, Geometry and Topology of the University of Valladolid. E-mail: tena@cpd.uva.es

I. ABSTRACT

A secret key variant of the the McEliece public key cryptosystem [4] was introduced by Rao and Nam in [1]. In this variant the matrix $G' = SGP$ is also kept secret. The system is vulnerable to Majority Voting analysis when the set of error vectors used to encrypt the message is restricted to those of weight at most $(d-1)/2$, but Majority Voting is not successful when the n-bit error vectors have average Hamming weight $n/2$ (See [2]). Therefore, Rao and Nam private key Cryptosystem uses a set $\mathcal{Z}$ of predefined error vectors with distinct syndromes and average Hamming weight $n/2$.

The main drawback of this system is that it needs to keep in memory the set of error vectors and syndromes so that the errors can be removed in the decryption process. This table of errors and syndromes must be as big as possible because a small cardinality of the set of predefined error vectors introduce weakness in the cryptosystem.

The objective of the Rao-Nam secret key was to use simple and small codes that require less storage and a simpler process than in the McEliece public key cryptosystem. The analysis of the system made necessary to increase the size of the codes (using lengths of 250 bits) in order to make the system stronger.

Our proposal is to use product codes. Let us recall that a product code is the set of all the matrices whose columns belong to a column code C_1 and whose rows are in a row code C_2. The parameters are $n = n_1n_2$, $k = k_1k_2$ and $d = d_1d_2$. (See [3]) The minimum distance is quite poor when compared with the length, but when the errors are in certain patterns we can take advantage of the structure of the product code to correct many more errors than the correcting capability of the code. To be precise in this work we make use of the erasure correcting capability of the factor codes in order to correct error vectors of average Hamming weight $n/2$. This presents two main advantages:

- There is no memory needed to store tables of syndromes and error vectors since the error vectors used come from $n_1 \times n_2$ matrices arbitrarily chosen among those whose support satisfies certain conditions. On the other hand, those errors can be corrected by using the erasure correcting capabilities of the factor codes.
- The decryption process is easy since it is done by decoding the factor codes which are of much smaller size.

Acknowledgements

This work has been partially supported by a project from Junta de Castilla-León.

References

[1] T.R.N. Rao and K.H. Nam. "Private-Key Algebraic-Coded Criptosystems".*Lecture Notes on Computer Science.* vol.263, Springer Verlag 1987. pp.35-48.

[2] J.van Tilburg. "Security-Analysis of a Class of Cryptosystems Based on Linear Error-Correcting Codes". *Royal PTT* 1994, Nederlands.

[3] F.J. MacWilliams and N.J. Sloane. "The Theory of Error-Correcting Codes". North Holland Publishing Co. 1977.

[4] R.J. McEliece. "A Public-Key Cryptosystem Based on Algebraic Coding Theory". *DSN Progress Report.* 42-44. Jet Propulsion Laboratory 1978, California Inst. of Tech. Pasadena, Cal.

Cryptanalysis of 'less short' RSA Secret Exponents

Eric R. Verheul and Henk C.A. van Tilborg

Department of the Interior
P.O. Box 20010
2500 EA, the Hague, the Netherlands
E-mail e.verheul@ngi.nl
Department of Mathematics and Computing Science
Eindhoven University of Technology
5600 MB, Eindhoven, the Netherlands
E-mail henkvt@win.tue.nl

THE RSA system can be described by the modulus n being the product of two (large) primes p, q, and by the public and secret exponents e and d which are related by $e \cdot d \equiv 1 \pmod{\operatorname{lcm}(p-1, q-1)}$. The public exponent e and the modulus n are made public; the remaining parameters are kept secret. In a *typical* RSA system one has that $\gcd(p-1, q-1)$ is small, $e < n$, and p and q have approximately the same number of bits.

In some applications of RSA, it is desirable to have a short secret exponent d, as this reduces the execution times. For instance when RSA is used on a smart card to sign a document. In [1], an attack on a typical RSA system with a "small" secret exponent is described. This attack will give the secret exponent d (as well as the prime-factors of n), provided that the number of bits in d does not exceed (approximately) one-quarter of the number of bits in n.

One of the intriguing aspects of this attack is that it does not only make use of knowledge of the modulus n. Indeed, it also highly depends on information obtained from the public exponent e. So in this situation the problem of breaking RSA is essentially different from the problem of the factorization of n where only information on n is available..

Here, we describe an extension of Wiener's attack. It will turn out that we always obtain a substantial amount of "secret" information from n and e in a typical RSA system, i.e. when d is not small. As in [1], our attack is based on the theory of continued fractions.

Let K be defined by

$$e \cdot d = 1 + K \cdot \operatorname{lcm}(p-1, q-1) = 1 + \frac{K}{G}(p-1)(q-1),$$

where $G = \gcd(p-1, q-1)$. Further let k/dg be the reduced representation of the fraction K/dG,

Theorem 1 *The first j terms of the Continued Fraction-representation (and convergents) of k/dg can be determined, where j is such that the number of bits in the denominator of the j-th convergent is approximately one quarter of the number of bits in n.*

In particular, if $d < n^{1/4}$ it can be determined completely. This is the result of Wiener [1].

To estimate the complexity of our method for $d > n^{1/4}$, we define

$$\ln_2 d = \ln_2 n^{1/4} \; + \; r.$$

Theorem 2 *The uncertainty about k/dg and thus about d, p, and q is about $2r + 6$ bits.*

REFERENCES

[1] M.J. Wiener, *Cryptanalysis of Short RSA Secret Exponents*, IEEE Transactions on Information Theory, IT–36, May 1990, pp. 553-558.

Cyclic Codes and Permutations Suitable For DES-like Cryptosystems

Claude Carlet[1] , Pascale Charpin[2] and Victor Zinoviev[3]
[1]GREYC, Université de Caen, and INRIA-Rocquencourt, France
[2] INRIA, Domaine de Voluceau-Rocquencourt, BP 105 - 78153, Le Chesnay Cedex, France
[3] Institute for Problems of Information Transmission, Bol'shoi Karetnyi 19, GSP-4 Moscow 101447, Russia

Abstract — **Almost bent mappings oppose an optimum resistance to linear and differential cryptanalysis. Our first purpose is to develop the "coding theory" point of view for studying the existence of almost bent mappings. Moreover we want to give an overview of the most recent results and describe precisely the open problems. We also give new characterizations of almost bent mappings.**

I. Introduction

We denote by V_m the vector-space $GF(2)^m$, which will be often identified to the Galois field $GF(2^m)$. The expression:

$$\mu_F(a,b) = \sum_{x \in V_m} (-1)^{b \cdot F(x)+a \cdot x} \qquad (1)$$

where F is a mapping from V_m to V_m, a and b are elements of V_m and "." is the usual dot product on V_m, plays a role in several topics of information theory:

- sequences (e.g. m-sequences, cf. [7]);
- correlation-immune and resilient functions (cf. [3]);
- permutations suitable for block ciphers (cf. [5]).

We focus in this paper on the study of a special class of mappings that we define now:

Definition 1 *The mapping F is said to be an* almost bent *(AB) mapping if and only if $\mu_F(a,b)$ takes the values* 0 *and* $\pm 2^{\frac{m+1}{2}}$ *only, when $a \in V_m$ and $b \in V_m^* = V_m \setminus \{0\}$. AB mappings exist only when m is odd.*

The problem of determining AB mappings is a central problem in the study of permutations suitable for DES-like cryptosystems. A mapping F opposes an optimum resistance to linear cryptanalysis when the maximum of the absolute value of $\mu_F(a,b)$ reaches the lower bound $2^{\frac{m+1}{2}}$ [11]. Chabaud and Vaudenay proved in [5] that these mappings have the property that $\mu_F(a,b)$ takes the values 0 and $\pm 2^{\frac{m+1}{2}}$ only.

On the other hand, any AB mapping is *almost perfect nonlinear* (APN), i.e. opposes an optimum resistance to differential cryptanalysis (cf. [1][12]):

Definition 2 *The mapping F is APN if and only if for any nonzero vector a and any vector b, the equation $F(x)+F(x+a) = b$ admits at most two solutions in V_m.*

Recall that Nyberg gave two examples of APN mappings which are permutations on $GF(2^m)$: $x \rightarrow x^{2^i+1}$, i prime to m, and $x \rightarrow x^{2^m-2}$ [12]. It was proved later that the first one of these mappings is AB and the other is not[5].

Actually all these results are known in coding theory. Moreover a second example of AB permutations was known already, due to Kasami [9, p.379]. That is $x \rightarrow x^{2^{2j}-2^j+1}$, for any j such that $gcd(j,m) = 1$. Since the work of Kasami, no other infinite class of AB mappings was discovered.

In Section 2 we first establish the link between the problem of finding APN (and AB) mappings and the study of some linear codes. We explain how the primitive cyclic codes appear in this context and how their parameters could be used. The results on the quadratic case are based on the work of Kasami.

In Section 3, we give new results on the AB mappings. Considering these mappings as vectorial Boolean functions, we give an upper bound for their degrees. We give new characterizations related with some bent functions.

II. The coding theory point of view

Let $n = 2^m - 1$ (m odd) and denote by α a primitive nth root of unity.

Theorem 1 *Let F be a mapping on $GF(2^m)$ such that $F(0) =$ 0. Let us denote by C_F the linear binary code of length n defined by the parity check matrix*

$$\mathcal{H}_F = \left\{ \begin{array}{ccccc} 1 & \alpha & \alpha^2 & \ldots & \alpha^{n-1} \\ F(1) & F(\alpha) & F(\alpha^2) & \ldots & F(\alpha^{n-1}) \end{array} \right\}$$

where each entry is viewed as a binary vector of dimension m. The dual code is denoted by $(C_F)^\perp$.
Then we have:

- *(i) the mapping F is APN if and only if the code C_F has minimum weight at least five.*
- *(ii) the mapping F is AB if and only if the weights of the non zero codewords of the code $(C_F)^\perp$ form the following set:* $\{\ 2^{m-1},\ 2^{m-1} \pm 2^{(m-1)/2}\ \}$.

Corollary 1 *If the mapping F is AB, then the dimension of the code $(C_F)^\perp$ equals $2m$, i.e. the code C_F has dimension $2^m -$ $2m - 1$.*

From now on we consider that $F(X)$ is a polynomial of degree less than n with coefficients in $GF(2^m)$, denoted by

$$F(X) = \sum_{j=1}^{n-1} \delta_j X^j \ , \quad \delta_j \in GF(2^m)\ . \qquad (2)$$

Denote by I_F the set of exponents j such that $\delta_j \neq 0$. Set

$$T_F = \{\ cl(1)\ \} \cup \{\ cl(j) \mid j \in I_F\ \}\ ,$$

where $cl(j)$ is the 2-cyclotomic coset of j modulo n.

Theorem 2 *Assume that m is not a prime. Let g be any integer dividing m. Set*

$$\Lambda_g = \{\ t \in [1, 2^m - 2] \mid t \equiv 2^\ell \pmod{2^g - 1},\ \ell < g\ \}\ .$$

If T_F is contained in Λ_g then F is not APN.

A. The quadratic case

The polynomial $F(X)$ is said to be *quadratic* when $I_F \subseteq \mathcal{J}$,

$$\mathcal{J} = \{\ 2^k + 2^\ell \mid k \text{ and } \ell \text{ in } [0, m-1]\ \}\ .$$

Note that the associated Boolean function, $f(X) = \beta \cdot F(X) + \gamma \cdot X$, is also quadratic. *Then the code $(C_F)^\perp$ is contained in $\mathcal{R}(2,m)^*$, the punctured Reed-Muller code of order two.*

Theorem 3 *The mapping F, given by (2), is such that the dimension of $(C_F)^\perp$ equals $2m$; assume that F is quadratic. Then F is AB if and only if it is APN.*

Corollary 2 *Assume that F is a quadratic polynomial, written as follows: $F(X) = \sum_{j \in I} \delta_j X^j$, $I \subset \mathcal{J}$. Then F is AB if and only if it satisfies*

$$\sum_{j \in I, j = 2^s + 2^\ell, s > \ell} \delta_j\ \alpha^{jk}(\nu^{2^s} + \nu^{2^\ell}) \neq 0\ ,$$

for all $k \in [1, n-1]$ and for all $\nu \in GF(2^m) \setminus \{0,1\}$.

B. Cyclic codes with two zeros

In this paragraph we suppose that $F(X)$ is a power polynomial $X \longmapsto X^t$ where generally $gcd(t,n) = 1$, $n = 2^m - 1$. So the code C_F is the binary cyclic code whose zeros are α and α^t. More generally, we consider binary cyclic codes with two zeros $\{\alpha^r,\ \alpha^s\}$. Such a code $C_{r,s}$ has parity check matrix:

$$\mathcal{H}_{r,s} = \left\{ \begin{array}{cccccc} 1 & \alpha^r & \alpha^{2r} & \ldots & \alpha^{(n-1)r} \\ 1 & \alpha^s & \alpha^{2s} & \ldots & \alpha^{(n-1)s} \end{array} \right\}$$

The codes $C_{r,s}$ have minimum distance 2, 3, 4 or 5. When it is 5, the code defines an APN mapping. There exists a conjecture due to Welch: *the code $C_{1,t}$, with $t = 2^i + 3$, $i = \frac{m-1}{2}$, corresponds to an AB mapping.* New results on the distances 3 and 4 were recently obtained in [6] and [8]. We prove more results on distance 4. For instance:

1. *Consider the binary cyclic code $C_{r,s}$, with $r = 2^i + 1$ and $s = 2^j + 1$, $i > j$. If $(i+j, m) = 1$ and $(i-j, m) = 1$ then the code $C_{r,s}$ has minimum distance at least four. Otherwise $C_{r,s}$ has minimum distance three.*

2. *Consider the binary cyclic code $C_{1,t}$, with $t = 2^i + 3$. If m is odd and $gcd(i,m) = 1$ then the code $C_{1,t}$ has minimum distance at least four. Otherwise $C_{1,t}$ has minimum distance three.*

III. On bent functions and AB mappings

We now give an upper bound for the degree of any AB mapping. On the other hand, the definition of AB mappings is close to that of bent functions. There is in fact a nice relationship between these two notions. Others properties are developed in [4].

Theorem 4 *If F is an AB mapping, then the degree of F is less than or equal to $(m+1)/2$.*

Example of an AB permutation of highest degree: The inverse of the permutation $x \rightarrow x^3$ is: $x \rightarrow x^{\frac{2^{m+1}-1}{3}}$, since we have: $2^{m+1} - 1 = 2(2^m - 1) + 1$. The exponent $\frac{2^{m+1}-1}{3} = 1 + 2^2 + \ldots + 2^{m-1}$ has 2-weight $\frac{m+1}{2}$. Thus, the degree of the AB permutation $x^{\frac{2^{m+1}-1}{3}}$ is $\frac{m+1}{2}$.

For any mapping F from V_m to itself, we denote by δ_F the integral-valued function on $(V_m)^2$ whose value at (a,b) is the number of solutions in V_m of the equation $F(x) + F(x+a) = b$. We denote by γ_F the Boolean function on $(V_m)^2$ whose value at (a,b) is 1 if $a \neq 0$ and $\delta_F(a,b) \neq 0$, and 0 otherwise.

Theorem 5 *Let F be a mapping from V_m to itself.*

1. *F is APN if and only if the Boolean function γ_F has weight $2^{2m-1} - 2^{m-1}$.*
2. *F is AB if and only if γ_F is bent.*

From Theorem 5 can be deduced again the fact that any quadratic APN permutation is AB. We have also a sufficient condition for a mapping to be AB, that is related once again to bent functions:

Theorem 6 *A sufficient condition for F to be AB is that, for any nonzero b in V_m, the Boolean function $b \cdot F(x)$ is the restriction to V_m of a bent function on V_{m+1}, i.e. there exists a Boolean function $f_b(x)$ on V_m such that the Boolean function: $(x,\epsilon) \rightarrow b \cdot F(x) + \epsilon f_b(x)$ is bent on $V_m \times GF(2)$.*

Every quadratic AB mapping satisfies this condition.
Notice that, if F is a power function on $GF(2^m)$, it is sufficient to check the existence of f_b for $b = 1$ only.

References

[1] E. Biham and A. Shamir *Differential cryptanalysis of DES-like cryptosystems*, Journal of Cryptology, Vol. 4 No. 1 (1991)

[2] A.R. Calderbank, G. McGuire, B. Poonen and M. Rubinstein, *On a conjecture of Helleseth regarding pairs of binary m-sequences*, IEEE Transactions on Information Theory, vol 42, 988-990 (1996)

[3] P. Camion and A. Canteaut, *Construction of t-resilient functions over a finite alphabet*, EUROCRYPT'96, Advances in Cryptology, Lecture Notes in Computer Science 1070, 283-293 (1996)

[4] C. Carlet, P. Charpin and V. Zinoviev, *Cyclic codes and permutations suitable for DES-like cryptosystems*, in preparation.

[5] F. Chabaud and S. Vaudenay, *Links between differential and linear cryptanalysis.* In Advances in Cryptology, EUROCRYPT'94, Lecture Notes in Computer Science, N. 950, pp. 356-365, Springer-Verlag, 1995.

[6] P. Charpin, A. Tietäväinen and V. Zinoviev, *On binary cyclic codes with $d = 3$*, to appear in "Problems of Information Transmission".

[7] T. Helleseth & P.V. Kumar, Sequences with low correlation, *to appear as one of the chapters in the Handbook of coding theory* edited by Brualdy, Huffman and Pless, to be published by Elsevier.

[8] H. Janwa, G. McGuire & R.M. Wilson, *Double-error-correcting codes and absolutely irreductible polynomials over GF(2)*, Journal of Algebra 178, 665-676 (1995).

[9] T. Kasami, *The weight enumerators for several classes of subcodes of the 2nd order binary Reed-Muller codes*, Information and Control 18, 369-394 (1971).

[10] F. J. Mac Williams and N. J. Sloane, *The theory of error-correcting codes*, Amsterdam, North Holland 1977.

[11] M. Matsui *Linear cryptanalysis method for DES cipher*, EUROCRYPT'93 Advances in Cryptography, Lecture Notes in Computer Science 765, p. 386-397 (1994)

[12] K. Nyberg *Differentially uniform mappings for cryptography*, EUROCRYPT'93 Advances in Cryptography, Lecture Notes in Computer Science 765, p. 55-64 (1994)

Highly Nonlinear t-Resilient Functions

Kaoru Kurosawa, Takashi Satoh and Kentaro Yamamoto
Dept. of Electrical and Electronic Engineering
Tokyo Institute of Technology
2-12-1 O-okayama, Meguro-ku, Tokyo 152, Japan
Email: kurosawa@ss.titech.ac.jp

Abstract — **This paper shows a method to design (n,m,t)-resilient functions with high nonlinearity. For fixed n input bits and m output bits, our method gives higher nonlinearity than the method of Zhang et al., while their method gives larger resiliency t than ours.**

I. Introduction

An n-input and m-output function $F(x_1,\ldots,x_n) = (f_1,\ldots,f_m)$ is called a (n,m,t)-resilient function if any function obtained from F by keeping any t input bits constant is uniformly distributed. Such functions find their applications in key renewal and in stream ciphers. It is known that there exists a linear (n,m,t)-resilient function if and only if there exists a linear $[n,m,t+1]$-code. On the other hand, the nonlinearity N_F of F is defined as a distance from the set of affine (linear) functions. Ding, Xiao and Shan showed the best affine approximation (BAA) attack against stream ciphers. Matsui showed the linear attack against DES. Therefore, (n,m,t)-resilient functions which possess high nonlinearity are desireable. Recently, Zhang and Zheng showed how to transform linear (n,m,t)-resilient functions into nonlinear ones[6].

This paper shows another way to design (n,m,t)-resilient functions with high nonlinearity. Our design method is extremely simple. For the same n and m, our method gives higher nonlinearity than [6] while the method of [6] gives larger resiliency than our method.

The proposed method provides a tradeoff between resiliency t and nonlinearity N_F such as follows. Fix n and m using an intermediate parameter l. If we choose a large l, then a small t and a large N_F are obtained. If we choose a small l, then a large t and a small N_F are obtained.

II. Preliminaries

Let $x=(x_1,\ldots,x_n)$. Let f be a function: $\{0,1\}^n \to \{0,1\}$. Then $f(x)$ is balanced if $\left|\{x \mid f(x)=0\}\right| = \left|\{x \mid f(x)=1\}\right| = 2^{n-1}$. Let F be a function: $\{0,1\}^n \to \{0,1\}^m$. Then $F(x)$ is uniformly distributed if $\left|\{x \mid F(x)=\beta\}\right| = 2^{n-m}$ for any $\beta \in \{0,1\}^m$.

Proposition 1 *[2, p. 370] $F(x) = (f_1(x),\ldots,f_m(x))$ is uniformly distributed if and only if all nonzero linear combinations of $f_1,\ldots,f_m$ are balanced.*

Definition 1 *$F(x_1,\ldots,x_n) = (f_1,\ldots,f_m)$ is a (n,m,t)-resilient function if any function obtained from F by keeping any t input bits constant is uniformly distributed.*

Corollary 1 *$F(x_1,\ldots,x_n) = (f_1,\ldots,f_m)$ is an (n,m,t)-resilient function if and only if all nonzero linear combinations of $f_1,\ldots,f_m$ are $(n,1,t)$-resilient functions.*

For two functions $f(x)$ and $g(x)$, define $d(f,g) \triangleq \left|\{x \mid f(x) \neq g(x)\}\right|$.

Definition 2 *The nonlinearity of f, denoted by N_f, is defined as*

$$N_f \triangleq \min_{a_0,\ldots,a_n} d(f(x), a_0 \oplus a_1x_1 \oplus \cdots \oplus a_nx_n) \ .$$

Proposition 2 *[3] $N_f \leq 2^{n-1} - 2^{n/2-1}$.*

Definition 3 *The nonlinearity of $F(x) = (f_1(x),\ldots,f_m(x))$ denoted by N_F, is defined as the minimum among the nonlinearities of all nonzero linear combinations of the component functions of F:*

$$N_F \triangleq \min_g\{N_g \mid g = \bigoplus_{j=1}^{m} c_jf_j, c_j \in \{0,1\}, (c_1,\ldots,c_m) \neq (0,\ldots,0)\}.$$

Definition 4 *For $f(x)$, define*

$$\widehat{f}(\omega_1,\ldots,\omega_n) \triangleq \sum_x (-1)^{f(x)}(-1)^{\omega_1x_1+\cdots+\omega_nx_n} \ .$$

$f(x)$ is a bent function if $\left|\widehat{f}(\omega_1,\ldots,\omega_n)\right| = 2^{n/2}$ for any $(\omega_1,\ldots,\omega_n)$.

Proposition 3 *[3] The equality of Proposition 2 is satisfied if and only if f is a bent function.*

Definition 5 *$F(x_1,\ldots,x_n) = (f_1,\ldots,f_m)$ is an (n,m)-bent function if all nonzero linear combinations of $f_1,\ldots,f_m$ are bent functions.*

Proposition 4 *[4] There exists an (n,m)-bent function if and only if $n \geq 2m$ and $n =$even.*

III. Highly Nonlinear t-Resilient Functions

Let φ be a function: $\{0,1\}^n \to \{0,1\}$. Let ψ be any function: $\{0,1\}^l \to \{0,1\}$. Let $x=(x_1,\ldots,x_n)$ and $y=(y_1,\ldots,y_l)$. Define $f(x,y) \triangleq \varphi(x) \oplus \psi(y)$.

Lemma 1 *If $\varphi(x)$ is a $(n,1,t)$-resilient function, then $f(x,y)$ is a $(n+l,1,t)$-resilient function.*

Proof. Fix t bits among $(x_1,\ldots,x_n,y_1,\ldots,y_l)$ arbitrarily. For simplicity, suppose that the fixed bits are

$$x_1=b_1,\ldots,x_h=b_h, y_1=b_{h+1},\ldots,y_{t-h}=b_t.$$

First, $\varphi(b_1,\ldots,b_h,x_{h+1},\ldots,x_n)$ is balanced because $\varphi(x)$ is t-resilient and $h \leq t$. Therefore, for any fixed values $c_1,\ldots,c_{l-t+h}$,

$$\varphi(b_1,\ldots,b_h,x_{h+1},\ldots,x_n) \oplus \psi(b_{h+1},\ldots,b_t,c_1,\ldots,c_{l-t+h})$$

is balanced. Hence,

$$\varphi(b_1,\ldots,b_h,x_{h+1},\ldots,x_n)\oplus\psi(b_{h+1},\ldots,b_t,y_{t-h+1},\ldots,y_l)$$

is balanced. This means that $\varphi(x)\oplus\psi(y)$ *is t-resilient.*

Lemma 2 *Let* $g(x)\triangleq\varphi(x)\oplus 1$. *Then* $N_g = N_\varphi$.

Lemma 3 *The nonlinearity of* $f(x,y)$ *satisfies* $N_f \geq 2^l N_\varphi$.

Theorem 1 *For any even* l *such that* $l \geq 2m$, *if there exists a* $(n-l,m,t)$*-resilient function* $\Phi(x)$, *then there exists a* (n,m,t)*-resilient function* $F(x,y)$ *whose nonlinearity satisfies* $N_F \geq 2^{n-1}-2^{n-l/2-1}$.

Proof. Let the $(n-l,m,t)$*-resilient function be*

$$\Phi(x)=\Big(\varphi_1(x),\ldots,\varphi_m(x)\Big)\ .$$

On the other hand, from Proposition 4, there exists a (l,m)*-bent function*

$$B(y)=\Big(b_1(y),\ldots,b_m(y)\Big)\ .$$

Define

$$F(x,y)\triangleq\Big(\varphi_1(x)\oplus b_1(y),\ldots,\varphi_m(x)\oplus b_m(y)\Big)\ .$$

Now for any $(c_1,\ldots,c_m)\neq(0,\ldots,0)$, *let*

$$\begin{aligned} f(x,y) &\triangleq c_1(\varphi_1(x)\oplus b_1(y))\oplus\cdots\oplus c_m(\varphi_m(x)\oplus b_m(y)) \\ &= (c_1\varphi_1(x)\oplus\cdots c_m\varphi_m(x))\oplus(c_1b_1(x)\oplus\cdots c_mb_m(x)) \end{aligned}$$

From Corollary 1, $c_1\varphi_1(x)\oplus\cdots\oplus c_m\varphi_m(x)$ *is t-resilient. From Def. 5,* $c_1b_1(x)\oplus\cdots c_mb_m(x)$ *is a bent function. Then from Lemmas 1 and 3,* $f(x,y)$ *is t-resilient and* $N_f \geq 2^{n-l}(2^{l-1}-2^{l/2-1})$. *Therefore,* $F(x,y)$ *is a* $(n+l,m,t)$*-resilient function and* $N_F \geq 2^{n-1}-2^{n-l/2-1}$.

IV. Comparison

Zhang and Zheng showed how to transform linear resilient functions into nonlinear resilient functions.

Proposition 5 *[6] Let F be a linear* (n,m,t)*-resilient function and G be a permutation on* $\{0,1\}^m$ *whose nonlinearity is* N_G. *Then* $\widehat{F}=G\circ F$ *is a* (n,m,t)*-resilient function whose nonlinearity satisfies* $N_{\widehat{F}}=2^{n-m}N_G$.

This section shows that for the same n and m,

- Theorem 1 gives higher nonlinearity than Proposition 5.
- Proposition 5 gives larger resiliency than Theorem 1.

Suppose that we obtain a (n,m,t)-resilient function F with nonlinearity N_F from Theorem 1 and a $(n,m,\widehat{t})$-resilient function $\widehat{F}$ with nonlinearity $N_{\widehat{F}}$ from Proposition 5.

Theorem 1 requires the existence of a $(n-l,m,t)$-resilient function such that $l \geq 2m$. Proposition 5 requires the existence of a linear $(n,m,\widehat{t})$-resilient function. Therefore, if we ignore "linear", then $\widehat{t}\geq t$.

In Proposition 5, $N_G \leq 2^{m-1}-2^{m/2-1}$ from Proposition 2 and Def. 3. Therefore,

$$N_{\widehat{F}} \leq 2^{n-1}-2^{n-m/2-1}\ . \qquad (1)$$

On the other hand, from Theorem 1, $N_F \geq 2^{n-1}-2^{n-l/2-1} \geq 2^{n-1}-2^{n-m-1}$ since $l \geq 2m$. Hence,

$$N_{\widehat{F}} \leq 2^{n-1}-2^{n-m/2-1} < 2^{n-1}-2^{n-m-1} \leq N_F\ .$$

It is known that there exists a linear (n,m,t)-resilient function if and only if there exists a linear $[n,m,t+1]$-code. From [5], there exists a linear $[18,8,6]$-code. So there exists a linear $(18,8,5)$-resilient function. In Theorem 1, let $l=18$. Then we obtain a linear $(36,8,5)$-resilient function with nonlinearity

$$N_F \geq 2^{35}-2^{26}\ .$$

On the other hand, there exists a linear $[36,8,16]$-code from [1]. So there exists a linear $(36,8,15)$-resilient function. Then from Proposition 5 and (1), we obtain a linear $(36,8,15)$-resilient function with nonlinearity

$$N_{\widehat{F}} \geq 2^{35}-2^{31}\ .$$

We summarize the above results in the following table.

	Theorem 1	Proposition 5
t	5	15
N_F	$\geq 2^{35}-2^{26}$	$\leq 2^{35}-2^{31}$

Table 1: Comparison between Theorem 1 and Proposition 5 on $(36,8,t)$-resilient functions

V. Tradeoff between resiliency and nonlinearity

In Theorem 1, fix n and m. Then we can choose even l arbitrarily in $2m \leq l \leq n-m$. If l is large, then we obtain small t and large N_F. If l is small, then we obtain large t and small N_F. This tradeoff is illustrated in the following table for $n=36$ and $m=8$.

t	7	5	4	3	2	1	0
N_F-2^{35}	-2^{27}	-2^{26}	-2^{25}	-2^{24}	-2^{23}	-2^{22}	-2^{21}
l	16	18	20	22	24	26	28

Table 2: Tradeoff between t and N_F on $(36,8,t)$-resilient functions

References

[1] A. E. Brouwer, "Bounds on the minimum distance of binary linear codes,"http,//www.win.tue.nl/win/math/dw/voorlincod.html.

[2] S. Lidl, H. Niederreiter, "Finite Fields"; Encyclopedia of Mathematics and Its Applications 20, Cambridge University Press, 1983.

[3] W. Meier and O. Staffelbach, "Nonlinearity criteria for cryptographic functions," Advances in Cryptology — EUROCRYPT '89 Proceedings, Lecture Notes in Computer Science 434, Springer-Verlag, pp. 549–562, 1990.

[4] K. Nyberg, "Perfect nonlinear S-boxes," Advances in Cryptology — EUROCRYPT '91 Proceedings, Lecture Notes in Computer Science 547, Springer-Verlag, pp. 378–386, 1991.

[5] T. Verhoeff, "An updated table of minimum-distance bounds for binary linear codes," IEEE Trans. Inform. Theory, vol. IT-33, no. 5, pp. 665–680, 1987.

[6] X. M. Zhang and Y. Zheng, "On nonlinear resilient functions," Advances in Cryptology — EUROCRYPT '95 Proceedings, Lecture Notes in Computer Science 921, Springer-Verlag, pp. 274–288, 1995.

Steganography - Applications of Coding Theory

A.Z.Tirkel[1], C.F.Osborne, T.E.Hall[2]
Department of Physics
Monash University
Clayton 3168, Australia
Email: Andrew.Tirkel@sci.monash.edu.au

Abstract — **This paper is concerned with the generation of two and three dimensional patterns suitable for embedding watermarks on images and multimedia video. Binary, greyscale and color patterns are examined. New constructions are presented and analyzed.**

I. Introduction

THE proliferation of internet services and advances in computer technology have provided the motive for and the means of implementing document protection. Coding theory has been adapted to perform this and other tasks, including image compression. Coding theory has been successfully applied to communications, where one dimensional signal processing is involved. Images and multimedia graphics require two and three dimensional arrays. The third dimension may be time. Applications of such arrays include coded aperture imaging, structured light and steganography. The objective of steganography is to embed an imperceptible message, such as a copyright ”watermark” in an image array. Most methods employ slight randomness inherent in or tolerable in a natural image. These techniques, reviewed in [1] resemble spread spectrum.

II. Unique 2D and 3D Patterns with window properties

The method first described in [2] is the only one which uses spatial correlation for message recovery. Although significant advances in this technique have occurred since 1993, the issues of image registration and distortion compensation remain unsolved. Images can be distorted during transmission or deliberately or unintentionally by the recipient. Distortions include compression, cropping, rotation, skew, greyscale and color translations, rescaling, frame reordering etc. Image registration is a multidimensional analogue of synchronization in spread spectrum. Unique arrays with desirable properties, similar to those of m-sequences and GMW sequences are required. We focus on patterns with unique location and correlation properties. These patterns are compatible with standard image processing format. For non-correlative message recovery, the window property is usually sufficient. De Bruijn cycles [3] are one-dimensional examples. Two dimensional perfect maps can be constructed from a set of shifted de Bruijn cycles, such that the relative shifts also form a de Bruijn cycle. The wraparound must provide the last valid shift in the sequence. Alternative constructions use a small perfect map, not derivable from one dimensional de Bruijn cycles, such as $[2 \times 2 : 4 \times 4]$ to generate progressively larger maps [4]. Three dimensional maps can be constructed as a collection of shifts of the two-dimensional perfect map [5]. The wraparound constraint is less restrictive in this case. 2d and 3d perfect maps compatible with JPEG and MPEG processing will be presented. Relaxing the window property to exclude the all zeros window permits the construction of many arrays, including m-arrays and m-volumes. Further relaxation of the window constraint results in new square and balanced arrays. Details of their construction and decoding techniques will be presented. The extension of these arrays to non-binary characters is straightforward and will be demonstrated. This weak window property does not restrict the ability for unambiguous location, since no window appears more than once.

III. Arrays for Correlative Recovery

Correlative message recovery requires the embedded array to possess the following properties, in order of diminishing importance: (1) High Figure of Merit for periodic/aperiodic Autocorrelation (2) Balance or approximate balance (3) Window property (4) Compatibility with size and aspect ratio required by standard image compression (5) Random appearance. Property (1) is required for minimum ambiguity, (2) for minimum cross-correlation with the image, (3) for location of cropped, lost or corrupted data. (4) and (5) are self-evident. The aperiodic autocorrelation [6] is appropriate for isolated patterns. For periodic mosaics of many patterns the even and odd periodic autocorrelation become valid approximations, since edge effects become less important. Other array features such as alphabet compatibility and energy efficciency will be discussed. Table 1 presents a qualitative performance tradeoff between suitable 2d array constructions. PBA (Perfect Binary Arrays) are based on difference sets. PM (Perfect Maps) are two dimensional analogues of de Bruijn cycles. M-Arrays [7] are m-sequences of composite length folded into arrays. Where resistance to cryptographic attack is required m-arrays can be transformed into GMW arrays which maintain the autocorrelation property. We introduce new constructions called PA (Prime Arrays), based on m-sequences of prime length or on Legendre sequences. Examples of these are shown in Table 2 and Table 3. The autocorrelation of both arrays is excellent. The number of $p \times p$ Legendre constructions is $(p-1)$ and the crosscorrelation between them is $-p$, 0 or $+p$. For comparison, the autocorrelation peak is $p(p-1)$ and the sidelobes are 0 and $-p$. If required, the Legendre construction

Name	1	2	3	4	5
PBA	Perfect	Poor	Poor	Yes	Yes
PM	Poor	Perfect	Perfect	Yes	Yes
M-Array	Excellent	Excellent	Yes	Poor	Poor
Prime Array	Excellent	Yes	Yes	Yes	Poor

Tab. 1: 2d Array Performance

[2] Also at Scientific Technology

[2] Department of Mathematics